

Regulace teploty systému pomocí GSM sítě

Bc. Josef Bureš

Diplomová práce
2006



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky
akademický rok: 2005/2006

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Josef BUREŠ**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Automatické řízení a informatika**

Téma práce: **Regulace teploty systému pomocí GSM sítě**

Zásady pro vypracování:

- 1. Seznamte se s problematikou vzdáleného monitorování a řízení teploty systému pomocí mobilní sítě.**
- 2. Navrhněte monitorovací a řídicí soustavu, pomocí které bude možné vzdáleně řídit teplotu soustavy.**
- 3. Monitorování a řízení soustavy rozšiřte o přístup přes TCP/IP síť.**
- 4. Činnost navrženého systému ověřte na jednoduché reálné soustavě.**

Rozsah práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. CHUDÝ, V., PALENČÁR, R.: Meranie technických veličin. STU Bratislava, Bratislava, 1999
2. PUŽMAN, J.: Dálkový přenos dat. Praha, Grada, 1985
3. ZEŽULKA, F., FIEDLER, P.: Ethernet v průmyslové automatizaci. In: Sborník referátů z doprovodné konference Pragoregula 2000, Praha, 2000.
4. HORST, J.: Informační a telekomunikační technika. Praha, BEN, 2004.
5. SVOBODA, J.: Telekomunikační technika I, II. Praha, BEN, 2002.

Vedoucí diplomové práce: **Mgr. Milan Adámek, Ph.D.**
Ústav elektrotechniky a měření

Datum zadání diplomové práce: **14. února 2006**

Termín odevzdání diplomové práce: **26. května 2006**

Ve Zlíně dne 14. února 2006


prof. Ing. Vladimír Vašek, CSc.
pověřený děkan




prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Cílem této diplomové práce je navrhnout a realizovat technologické řešení regulace obecné tepelné soustavy s využitím mobilního zařízení GSM mobilní sítě jako vzdáleného ovladače. Samotný regulátor je založen na principu fuzzy regulace. Soustava je ovládána pomocí mobilního telefonu GSM sítě a dle potřeby také ovladatelná pomocí jednoduchého klient-server TCP/IP rozhraní, WWW rozhraní.

Regulátor, rozhraní klient-server a WWW server budou zavedeny do Embedded PC Advantech ® Biscuit PC. Jako operační systém je použit Microsoft ® Windows 95. WWW server je jednoduchý, volně šiřitelný WWW server s PHP interpreterem - Apache.

Výsledkem této diplomové práce je simulace tepelné soustavy, její regulace pomocí softwarového fuzzy regulátoru a ovládání a sledování regulačního procesu pomocí moderních komunikačních technologií.

Klíčová slova:

regulace, fuzzy regulátor, Advantech, GSM, WWW, PHP

ABSTRACT

The goal of this thesis is to design and to realize technological solution of the regulation process of the common thermal system with implemented GSM mobile device as a remote controller. This system is about to be controlled with a GSM network mobile phone and optionally with easy client-server TCP-IP interface and WWW interface too.

Regulator and client-server interface is loaded into Embedded PC Advantech ® Biscuit PC. As an operation system it is selected Microsoft Windows 95. WWW interface is integrated into the embedded PC. WWW server is open source, freeware server with PHP language interpreter – Apache.

Finally, result of this thesis is to simulate thermal system, its regulation by help of software-made fuzzy regulator and its control and monitoring using modern communication technologies.

Keywords:

regulation, fuzzy regulator, Advantech, GSM, WWW, PHP

PODĚKOVÁNÍ

Dovoluji si tímto poděkovat panu Mgr. Milanu Adámkovi, Ph.D., vedoucímu diplomové práce, za pomoc při sestavování mé diplomové práce, za usměrnění mé činnosti při plnění zadání diplomové práce a za všechny zodpovězené dotazy týkající se dané problematiky. Současně bych rád poděkoval jeho kolegům za pomoc při sestavování reálné regulované soustavy.

Poděkování patří také panu Ing. Viktoru Bradovi za odborné rady týkající se vytváření softwarového vybavení pro praktickou část mé práce a firmě Datasys s.r.o., která mě umožnila provést praktickou část s využitím jejího technologického vybavení.

Závěrem chci vyjádřit poděkování mé přítelkyni Heleně Strakové, rodině a přátelům za jejich duševní podporu a trpělivost.

Ve Zlíně, dne 28. srpna 2006

.....

Bc. Josef Bureš

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 GSM	12
1.1 VZNIK, VÝVOJ A UŽITÍ MOBILNÍ SÍTĚ GSM.....	12
1.1.1 Rok 1982 – Počátek GSM.....	13
1.1.2 Období 1986 - 1991	13
1.1.3 Vznik prvního operátora a počátky využívání GSM	14
1.1.4 Vývoj GSM v letech 1996 – 2000.....	15
1.1.5 Vývoj GSM od roku 2000 do současnosti	15
1.2 GPRS (GENERAL PACKET RADIO SERVICE).....	16
1.2.1 Funkcionalita a nabízené služby	16
1.2.2 Popis GPRS komunikace	17
1.3 EDGE (ENHANCED DATA FOR GSM EVOLUTION).....	20
1.4 SÍTĚ UMTS – MOBILNÍ SÍTĚ TŘETÍ GENERACE.....	22
1.4.1 Dělení sítí UMTS	22
1.4.2 Struktura sítě UMTS	22
1.4.3 Výhody UMTS.....	23
1.4.4 Služby UMTS FDD.....	23
1.5 SATELITNÍ MOBILNÍ SÍŤ IRIDIUM.....	24
2 POPIS EMBEDDED TECHNOLOGIE	25
2.1 EMBEDDED TECHNOLOGIE.....	25
2.2 HISTORICKÉ KOŘENY EMBEDDED TECHNOLOGIÍ – JEJÍ ZAČÁTKY V 60. LETECH MINULÉHO STOLETÍ.....	26
2.2.1 Vývoj elektroniky a embedded technologií v 70. letech minulého století.....	29
2.2.2 První 8 bitový mikroprocesor Intel 8080	30
2.2.3 Motorola HCF 6800	30
2.2.4 První víceúčelový mikrokontrolér Intel 8051	31
2.2.5 Počátek věku informačních technologií v 90. letech minulého století, nástup miniaturizace a hromadné využívání elektroniky v průmyslu.....	32
2.2.6 Vznik sběrnice PC/104.....	33
2.2.7 Universal Serial Bus - USB.....	34
2.2.8 Současná výpočetní technika	35
2.3 ARCHITEKTURA EMBEDDED SYSTÉMŮ.....	35
2.3.1 Dostupné platformy mikroprocesorů pro embedded systémy	35
2.3.2 RISC architektura.....	37
2.3.3 Operační systémy, vývojová prostředí.....	38
2.3.4 Ladění chodu embedded aplikací (debugging)	40
2.3.5 Postup a pravidla při psaní programů pro embedded systémy.....	40
2.3.6 Vytvoření systémové platformy, startovací rutiny a chod embedded systémů.....	41
2.3.7 Typy architektur embedded systémů	42

2.3.8	Příklady využití embedded systémů v reálném světě	46
2.3.9	Současní světoví výrobci embedded systémů, jejich příslušenství a softwarového vybavení.	46
3	FUZZY LOGIKA A FUZZY MNOŽINY	47
3.1	PRINCIPY FUZZY LOGIKY A FUZZY REGULACE	47
3.2	POUŽITÍ FUZZY LOGIKY PŘI REGULACI.....	47
3.3	FUZZY LOGIKA	48
3.4	VYUŽITÍ FUZZY LOGIKY.....	48
3.5	FUZZY REGULÁTORY	48
3.6	PRINCIP FUZZY REGULACE.....	48
3.6.1	Základní postup při aplikaci fuzzy regulace	49
3.6.2	Typy základních regulátorů a jejich fuzzy variant.....	49
3.6.3	Stanovení parametrů fuzzy regulátoru	51
3.7	FUZZY MODEL ŘÍZENÉHO PROCESU.....	52
II	PRAKTICKÁ ČÁST.....	53
4	NÁVRH MONITOROVACÍ A ŘÍDICÍ SOUSTAVY	54
4.1	VZNIK MONITOROVACÍ A ŘÍDICÍ SOUSTAVY.....	54
4.1.1	Podmínky pro realizaci	54
4.1.2	Shrnutí faktů a budoucí použitelnost modelu jako vzoru	54
4.2	POPIS TECHNOLOGICKÉHO PROCESU.....	55
4.2.1	Základní informace vybrané o tepelné soustavě.....	55
4.2.2	Popis prostředí tepelné soustavy	55
4.2.3	Použité topné těleso.....	57
4.3	NÁVRH MONITOROVACÍ A ŘÍDICÍ SOUSTAVY	57
4.3.1	Schéma soustavy a monitorovacího a řídicího systému.....	57
4.4	SCHÉMA ŘEŠENÍ	58
4.4.1	Popis schématu.....	58
5	POUŽITÝ HARDWARE.....	60
5.1	EMBEDDED PC ADVANTECH ® BISCUIT PC.....	60
5.1.1	Základní popis PCM-5822-D0A1	61
5.1.2	Parametry vstupně výstupních zařízení.....	61
5.1.3	Zobrazovací zařízení	62
5.1.4	Mechanické parametry a pracovní podmínky	62
5.1.5	Napájení	62
5.1.6	Rozložení součástí na základní desce	63
5.2	UŽIVATELSKÝ POČÍTAČ	63
5.2.1	Základní hardwarové parametry	63
5.2.2	Operační systém a instalovaný kancelářský a vývojový software	64
5.3	ŘÍDICÍ ČLEN.....	64
5.3.1	Relay desk v1.0	65
5.3.2	Com Switch v1.0.....	65
5.3.3	Teplotní čidlo TM fy PaPouch ™ Elektronika	66

5.3.4	Mobilní telefon Siemens M75.....	67
5.3.5	Datový sériový kabel DCA-500.....	68
6	POUŽITÝ SOFTWARE.....	69
6.1	OPERAČNÍ SYSTÉM V BISCUIT PC.....	69
6.2	APACHE WWW SERVER.....	70
6.3	VÝVOJOVÉ PROSTŘEDÍ	70
6.4	VLASTNÍ SOFTWARE - REGULÁTOR	71
6.4.1	Typ regulátoru.....	71
6.4.2	Konfigurace regulátoru	72
6.4.3	Stanovení souboru fuzzy výrazů	73
6.4.4	Stanovení souboru fuzzy hodnot.....	73
6.4.5	Základní struktura programu regulátoru	74
6.4.6	Hlavní funkce programu regulátoru	74
6.4.7	Objekt pro práci se sériovým portem.....	75
6.4.8	Objekt pro práci s paralelním portem.....	75
6.5	REGULÁTOR - SOUČÁST GSM SERVER	77
6.6	REGULÁTOR - SOUČÁST TELNET SERVER	77
6.7	AKTIVNÍ SERVEROVÝ SKRIPT PHP.....	78
7	REGULOVANÁ SOUSTAVA	79
7.1	VÝROBA PLOŠNÉHO SPOJE FOTO CESTOU.....	79
7.2	VÝROBA RELAY DESK V 1.0	80
7.2.1	Potřebné součástky.....	80
7.3	VÝROBA COM SWITCH V1.0.....	81
7.3.1	Potřebné součástky.....	81
7.4	SESTAVENÍ KORPUSU SIMULÁTORU REGULOVANÉ SOUSTAVY	82
7.5	INSTALACE ROZVODŮ ELEKTRICKÉ ENERGIE	83
7.5.1	Instalace rozvodů 230V.....	83
7.5.2	Instalace rozvodů nízkého napětí +5V a +12V	84
7.5.3	Kabeláž.....	84
7.6	ELEKTRICKÉ SPOTŘEBIČE	84
7.6.1	Montáž zdroje stejnosměrného napětí.....	85
7.6.2	Instalace topného tělesa	85
7.6.3	Montáž ventilátoru	85
7.6.4	Montáž reléové desky (Relay Desk v 1.0)	86
7.6.5	Montáž přepínače sériových portů (Com Switch v1.0)	87
7.6.6	Instalace Biscuit PC Advantech ® PCM-5822	87
8	POPIS STARTU, INICIALIZACE A PRACOVNÍHO CYKLU MONITOROVACÍHO A ŘÍDICÍHO SYSTÉMU.....	88

8.1	SPUŠTĚNÍ HARDWARE.....	88
8.2	INICIALIZACE SOFTWARE REGULÁTORU	88
8.3	PRACOVNÍ CYKLUS.....	89
9	FÁZE VZNIKU PROJEKTU A ROZPOČET	90
9.1	VYTVÁŘENÍ NÁVRHU.....	90
9.2	KONTROLA NÁVRHU.....	90
9.3	REALIZACE.....	91
9.4	TESTOVÁNÍ.....	91
9.5	ROZPOČET	92
10	OVĚŘENÍ ČINNOSTI ŘÍDICÍHO A MONITOROVACÍHO SYSTÉMU POMOCÍ REÁLNÉ REGULOVANÉ SOUSTAVY	93
10.1	FYZIKÁLNÍ ZÁKLADY MODELU REÁLNÉ TEPELNÉ SOUSTAVY.....	93
10.1.1	Základní informace o tepelné soustavě.....	93
10.1.2	Tepelně bilanční rovnice	96
10.1.3	Maximální dosažitelný rozdíl teplot dané tepelné soustavy	99
10.1.4	Čas potřebný pro zvýšení teploty soustavy o jeden stupeň.....	99
10.1.5	Čas potřebný pro zvýšení teploty o maximální rozdíl	100
10.2	TESTOVÁNÍ REGULÁTORU	100
10.2.1	Výpočet nejrychlejšího času pro dosažení zvýšení teploty soustavy o 7 °C.....	100
10.2.2	Testování regulátoru.....	101
10.2.3	Testování ventilace, úloha ventilace v systému	102
10.2.4	Zhodnocení výsledků testu.....	102
	ZÁVĚR.....	104
	SEZNAM POUŽITÉ LITERATURY.....	105
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	109
	SEZNAM OBRÁZKŮ	112
	SEZNAM TABULEK.....	114
	SEZNAM ROVNIC	115
	ABECEDNÍ REJSTŘÍK	116
	SEZNAM PŘÍLOH.....	120

ÚVOD

V současné době dochází v naší zemi k masivnímu využívání GSM technologií a to jak ve sféře privátní, tedy osobní mobilní zařízení – telefonní přístroje, inteligentní PDA, tak i ve sféře průmyslové, tedy industriální řešení na bázi GSM standardu.

Vývoj v posledních letech zaznamenal ohromný postup vpřed, GSM zařízení jsou ve stále větší míře miniaturizována, jejich výdrž díky vysoce kvalitním přenosným zdrojům energie je na vysoké úrovni a kvalita poskytovaných služeb spojených s GSM stále roste. Tímto se tato technologie stává velmi významnou součástí našeho života.

Pominu-li fakt, že dnešní GSM zařízení bývají ve většině případů vybaveny kvalitním displejem, kvalitním ovládním, vysokou funkcionalitou a množstvím doplňků, jako jsou GPS®, Java™, jsou také mnohem flexibilnější, co se týče dostupnosti služeb. Mobilní operátoři každým dnem zvyšují kvalitu poskytovaných služeb a zvyšují kapacitu a dostupnost svých sítí. Hustota mobilních sítí na našem státním území je v současnosti téměř na 98% pokrytí. Pokrytí se samozřejmě zlepšuje v hojnější míře ve větších městských aglomeracích, v Praze je dokonce i v metru. Na vesnicích si lidé také na bezdrátové telefonování zvykli. Za oblíbenost této technologie mluví i ten fakt, že jsou v naší republice v současné době tři mobilní operátoři, kteří mají dohromady přes 5 milionů aktivních čísel. Navíc

cenová politika mobilních operátorů směřuje k nižším hladinám s větší nabídkou služeb.

Výše uvedená fakta tedy jednoznačně podporují vznik velkého množství možných aplikačních programů, které tuto technologii využívají.

Já se ve své práci zabývám jedním z mnoha možných využití sítě GSM – jedná se o návrh dálkového ovládní tepelné soustavy. Tato koncepce se zdá být velmi přínosná do budoucna, neboť požadavků na vzdálenou regulaci bude jistě přibývat. A co je výhodnější jako prostředek komunikace než-li kvalitní, laciná bezdrátová síť.

Cílem této práce je dokázat, že tato technologie je k tomuto účelu nejen použitelná, ale že její nasazení je v mnohých případech dokonce nezbytné. Příkladem takového systému může být například regulace vytápění rodinného domu, bytu nebo také chaty, což už jistě osloví většinu z nás.

I. TEORETICKÁ ČÁST

1 GSM

1.1 Vznik, vývoj a užití mobilní sítě GSM

Global System for Mobile communication - Globální systém pro mobilní komunikace - je celosvětový standard pro digitální celulární telefonickou službu. Sítě GSM byly postaveny jako alternativa stávajících systémů AMPS („Advanced Mobile Phone Service“) a v budoucnosti budou podporovat pokročilé datové aplikace. Standard GSM je určen pro evropské trhy (ale v současnosti je k dispozici také v USA), kde nabízí automatický, mezinárodní roaming ve více zemích.

Mobilní telefonní síť GSM se skládá z mnoha důležitých částí. Každá mobilní síť má řídicí centrum. Zde jsou umístěny všechny důležité prvky, které se starají o celý běh sítě. Je to například středisko textových zpráv, nebo podpůrné centrum, které se stará o takové záležitosti jako je účtování hovorů nebo sledování činnosti celé sítě. Podobně důležitou částí jsou vysílače, které pokrývají území a zajišťují rádiové spojení sítě s telefonem.

Území, které je pokryto signálem mobilní sítě, je rozděleno na malé oblasti, které obsluhuje jeden vysílač: říkáme jim buňky. Jedna buňka by se dala přirovnat k obci a je to vlastně nejmenší samosprávná jednotka, kterou obsluhuje právě jeden vysílač. Tomu se v žargonu techniků říká BTS (je to zkratka z anglického „Base Transceiver Station“). U nás čítá síť každého z našich tří operátorů několik tisíc takových vysílačů a buněk.

Pět až patnáct vysílačů je ovládáno jedním řídicím vysílačem (přesněji BSC z anglického „Base Station Controller“, kontrolér základnových stanic). Aby veškerá komunikace nemusela probíhat v celé síti, řídicí vysílače se slučují do celků, kterým se říká lokální oblasti - LA (anglicky „Local Area“).

Dole na stožáru mobilní sítě je skříň, ve které se nachází napájecí systémy vysílače. V horní části vysílače jsou antény, které komunikují s mobilem: je to buď několik směrových antén, které vysílají jen určitým směrem, nebo jedna všesměrová. Kromě zajištění komunikace stožáru s mobilem je ale důležité, aby byl stožár zapojen do mobilní sítě. Proto jsou kromě antén na stožáru upevněny i malé parabolické vysílače, tzv. mikrovlnné spoje. Vysílač je jimi propojen s ostatními vysílači nebo s řídicím vysílačem (BSC).

Každý stožár obsluhuje jednu buňku a má přiděleny konkrétní frekvence (kanály), na kterých smí vysílat. Tyto frekvence musí být odlišné od frekvencí, na kterých vysílají sousední buňky, aby se jejich signály vzájemně nerušily. [10]

Mobilní síť GSM pracuje v základních pásmech o frekvencích 900, 1800 a 1900 MHz. Reálných využitých frekvenčních pásem je ve skutečnosti více než sto, ale pro zjednodušení se používají právě tyto výše uvedené hlavní frekvence.

Na jeden kanál může být připojeno nanejvýš osm telefonů, na druhou stranu ale jeden vysílač může vysílat i na několika kanálech. U nás jsou většinou využívány dva až čtyři kanály pro jednu buňku a najednou tedy může telefonovat i přes třicet uživatelů. Aby bylo možné obsluhovat více než jeden telefon v jednom okamžiku, je každý kanál rozdělen na tzv. timesloty, neboli časové úseky – každý reprezentuje jeden hovor. Dohromady jich je osm, přičemž každá buňka vysílá ještě tzv. signalizační timeslot, který komunikuje se všemi telefony najednou – zajišťuje například sestavení hovoru nebo přenos textových zpráv.

Jednotlivé telefony se v komunikaci s vysílačem postupně střídají, přičemž každý z nich má možnost komunikovat jen velice krátkou dobu (asi 0,5 ms). Těchto cyklů s osmi kroky se za sekundu uskuteční více než dvě stě. [8]

1.1.1 Rok 1982 – Počátek GSM

Nordic Telecom a Nizozemský PTT navrhly Conference of European Post and Telecommunications (CEPT), aby vyvinul nový standard pro digitální celluární systémy, které by zahrnuly všechny požadavky na Evropské mobilní síť. Evropská komise vydala nařízení, které požaduje po členských zemích, aby rezervovaly frekvenci 900 Mhz pro GSM.

1.1.2 Období 1986 - 1991

V roce 1986 byly pečlivě vybrány techniky pro radiový přenos v GSM. O rok později 13 světových operátorů a administrátorů podepsalo v září chartu GSM (Groupe Speciale Mobile), která nabyla platnosti 1. července 1991. Originální Francouzský název byl později změněn na Global System for Mobile Communications, ale originální zkratka GSM zůstala. Byla navrhována specifikace GSM.

V roce 1989 Evropský institut telekomunikačních standardů (ETSI - „European Telecommunications Standards Institute“) definoval GSM jako mezinárodně přijatelný standard pro digitální celluární telefonii. Pro GSM vznikla v ETSI speciální komise.

V roce 1990 byla upravena Specifikace Phase 1, GSM 900 byla ustálena. Začaly úpravy pro „Digital Cordless Standard“ - DCS (GSM 1800). Dále byly implementovány „Validation system standards“ a konal se první světový kongres GSM v Římě, kterého se účastnilo 650 účastníků. O rok později byla představena první specifikace GSM, stávající specifikace DCS (1800) byla ustálena a konal se světový kongres GSM v Nice s celkem 650-ti účastníky.

1.1.3 Vznik prvního operátora a počátky využívání GSM

První funkčním GSM operátorem se 27. března 1991 stává Oy Radiolinja Ab ve Finsku a vzápětí v prosinci téhož roku existovalo v 7 zemích celkem 13 sítí. V témže roce se konal světový kongres GSM v Berlíně.

V roce 1993 bylo podepsáno několik prvních roamingových smluv - tedy smluv mezi národními GSM operátory o vzájemném poskytování telekomunikačních služeb. Téhož roku v prosinci bylo v 18 zemích celkem již 32 různých sítí. V portugalském Lisabonu se konal GSM kongres, kterého se účastnilo celkem 760 účastníků.

Převratem bylo o rok později umožnění přenášení dat a faxů přes GSM síť – stalo se tak v GSM síti Vodacon. V roce 1994 se opět konal každoroční kongres GSM, tentokrát v Athénách v Řecku. V prosinci téhož roku již GSM expandovalo do dalších 11 zemí a ke konci roku existovalo již 69 různých sítí.

Rozběhl se roaming v oblasti dat, faxu a SMS. Standard GSM Phase2 byla dokončena, včetně všech změn pro Personal Communications Services PCS 1900 (systém v USA). Americká Federální Komise pro Komunikace (FCC - Federal Communications Commission) vydražila licence na PCS (1900). Za krátký čas na to v USA vznikla první síť PCS 1900. Firma Nokia představila na veletrhu Telecom'95 v Ženevě přenos dat o rychlosti 33.6 kbps přes GSM. Telefon Ericsson 377 se stal nositelem označení Telefon roku.

Během roku 1995 se GSM formálně zaregistrovala jako asociace ve Švýcarsku. V den svého vzniku měla 156 členů z 86-ti různých zemí. Ve Španělsku se konal kongres a počet účastníků dosáhl čísla 1400. V prosinci roku 1995 bylo již 117 sítí v 69 zemích.

1.1.4 Vývoj GSM v letech 1996 – 2000

V prosinci roku 1996 bylo již 120 GSM sítí v 84 zemích. Světový kongres GSM se konal v Cannes a konal se zde pravidelně každý rok až do roku 2000. V průběhu roku proběhlo zasedání MoU („Memorandum of Understanding“) v Atlantě. Na trhu se objevily první SIM karty s pamětí 8192 byte. Operátoři GSM sítí zahájili prodej předplacených SIM karet. Firma Option International představila první GSM modem.

Během roku 1997 byly na oběžnou dráhu kolem Země vypuštěny satelity pro systém Iridium. Firma Bosh představila první duální telefon (900 – 1900 MHz). V průběhu roku 1998 GSM operátor Vodacom představil první VoiceMail (hlasová schránka) zdarma a zahájil služby Yebo!Net. Bohužel tento rok také došlo k jedné nemilé události, a to, že v USA byla nabourána SIM karta, tedy byla prolomena bezpečnost dat uvnitř. GSM 1900 oslavilo velký úspěch, protože služeb této sítě využívalo v roce 1998 přes 2 miliony uživatelů. Option International představilo FirstFone-PC Card II modem pro mobilní telefony GSM 900. Firma MTN odpověděla na výzvu Boshe a zavedla také VoiceMail zdarma a ještě navíc převod volných minut mezi účetními obdobími (CarryOver minutes). Satelitní GSM síť Iridium přešla do ostrého provozu. V Singapuru začaly pokusy s HSCSD (High-Speed Circuit-Switched Data). V prosinci bylo již 125 milionu uživatelů GSM (900/1800/1900)

V roce 1999 proběhly první pokusy s GPRS v USA a ve Skandinávii, první pokusy s WAP ve Francii a Itálii, GSM MoU se připojilo k Third Generation Partnership Project (3GPP), byly zprovozněny první sítě GPRS, byla uvolněna specifikace Bluetooth v 1.0. Na konci roku 1999 služeb GSM (900/1800/1900) využívalo již přes 165 milionu uživatelů. [9]

1.1.5 Vývoj GSM od roku 2000 do současnosti

V těchto letech došlo k ohromnému rozmachu informačních technologií a GSM samozřejmě prošel dalším vývojem. Jedna nevýhoda pro GSM ovšem přetrvávala v podstatě beze změn - úzká šířka přenosového pásma - bandwidth. Vývoj služeb se však rokem 2000 nezastavil. Operátoři se i nadále předháněli v inovacích. Dokonalejší mobilní telefony s barevnými displeji přímo vybízely k experimentům. Vznikla Multimediální zpráva -

MMS. Začala se psát nové éra. SMS však neztratila nic ze svého lesku, ba naopak. Krátké textové zprávy posíláme v stále hojnější míře.

1.2 GPRS (General Packet Radio Service)

GPRS (General Packet Radio Service) přenos dat pomocí paketů/Mikroprohlížeč. Mikroprohlížeč WAP 1.1 umožňuje uživateli přístup ke službám mobilního internetu, které jsou nabízeny poskytovateli služeb. Na dosah ruky máte finanční, sportovní a cestovní informace. Při cestování nyní můžete mít přístup k chatovacím programům, ke hrám a k zábavě – rychle a bezdrátově! GPRS uživatelům umožňuje rychlejší provádění hovorů, vyšší přenosové rychlosti a permanentní ("always on") připojení. - SMG31 prohlížeč WAP GPRS 1TX (data do systému), 3RX (data ze systému) kódování GPRS plná podpora mobilním zařízením Třídy B.

Specifikaci systému GPRS vypracovala organizace ETSI. Do testovacího provozu bylo GPRS uváděno zhruba od roku 1999 a v témže období docházelo také k podepisování prvních komerčních smluv s operátory. U nás GPRS spustil jako první Eurotel ve druhé polovině roku 2000. Po něm službu spustili postupně i T-Mobile a Oskar (nyní Vodafone).

1.2.1 Funkcionalita a nabízené služby

Datové přenosy byly v systému GSM možné už od jeho vzniku. Spojení s klasickým GSM je však stejně jako v ostatních radiotelefonních sítích založeno na přepínání okruhů (komutované spojování). Po vytvoření spojení je tedy po celou dobu hovoru využíván jeden virtuální spojovací kanál, který sice zajišťuje jistou požadovanou kvalitu služby, ale na druhou stranu nevyužívá příliš efektivně kapacitu daného přenosového systému. Kanál je totiž obsazen i v případě, že oba koncoví účastníci nevyvíjejí žádnou aktivitu (mlčí). Se zvyšujícími se nároky na kapacitu přenosového pásma se stále častějším využíváním datových přenosů přes spojově orientovanou telefonní síť GSM (CSD – Circuit Switched Data, obecněji pak také DoV – Data Over Voice) rostla potřeba najít vhodnou technologii, která by s omezenou přenosovou kapacitou nakládala šetrněji. Přenosová rychlost 9,6 kbps, kterou nabízelo v základní verzi klasické GSM, navíc začala být nedostatečná. Řešení tohoto problému vzniklo postupně několik a jedním z nich je právě systém GPRS.

1.2.2 Popis GPRS komunikace

Systém GPRS je inspirován, jak už napovídá název, paketovým způsobem přenosu, který používají zejména technologie pevných datových sítí. Přenášená data jsou, zjednodušeně řečeno, na straně odesílatele nejprve rozdělena na bloky (pakety). Ty se označí, poté „zabalí“ do informací přenosových protokolů a odděleně pošlou k cíli. Podle momentálního stavu konkrétní přenosové sítě mohou dokonce související části dat přijít k cíli různými cestami. Na straně příjemce se datové bloky opět poskládají a mohou se zpracovávat.

U GPRS není třeba otevírat permanentní přenosový kanál, ale stačí jej otevřít až v případě potřeby „na požádání“ a vyhradit volné místo v kanálu pro přenos dat a data pak poslat směrem k adresátovi. Vyhrazení místa pro přenos dat spočívá v alokaci potřebného (dovoleného) počtu z celkových osmi timeslotů v daném rádiovém kanálu. Počet timeslotů, které lze vyhradit, závisí na třídě podporovaného GPRS. Po zbytek doby, kdy mezi účastníky sítě GPRS žádná data neputují, nejsou trvale zabrány žádné timesloty a kapacita kanálů je plně k dispozici ostatním. V klidovém stavu sice GPRS také potřebuje síťovou kapacitu, ale oproti aktivnímu stavu pouze v zanedbatelném množství.

Teoretická maximální přenosová rychlost fyzické vrstvy GPRS je stanovena na 171,2 kbps. To je ale pouze vykalkulovaná teorie, která počítá s využitím všech osmi timeslotů daného účastnického kanálu. V praxi se s takovou přenosovou rychlostí GPRS nesečkáme, neboť žádný operátor nepřístupí na alokaci celého kanálu jen pro jednoho účastníka a navíc je třeba ke kódovaným uživatelským datům před vlastním přenosem přidat ještě dodatečné informace použitých přenosových protokolů. Např. třída GPRS s číslem 10 umožňuje zabrat 4 + 2 timesloty (download + upload), ale v součtu vždy maximálně pět současně. Pro představu lze uživatelské přenosové rychlosti pro třídu GPRS (podle počtu alokovaných timeslotů) určit z předpokladu 6,7 až 16,7 kbps na jeden timeslot. Třídy GPRS

s možným přiděleným množstvím timeslotů zobrazuje tabulka č. 1.

Na přenosovou rychlost GPRS mají také vliv kódovací schémata CS (Coding Scheme) s různým zabezpečením proti chybám při přenosu rádiovým prostředím. Jsou celkem čtyři - CS1 až CS4 - a záleží na operátorech, která z kódovacích schémat zvolí pro použití ve své síti.

Popis služeb GPRS je přehledně zpracován v příloze P I.

Tab. č. 1 – Třídy GPRS zařízení

GPRS Class (Třída GPRS)	Timesloty downlink (k uživateli)	Timesloty uplink (od uživatele)	Timesloty současně aktivní
1	1	1	2
2	2	1	3
3	2	2	3
4	3	1	4
5	2	2	4
6	3	2	4
7	3	3	4
8	4	1	5
9	3	2	5
10	4	2	5
11	4	3	5
12	4	4	5

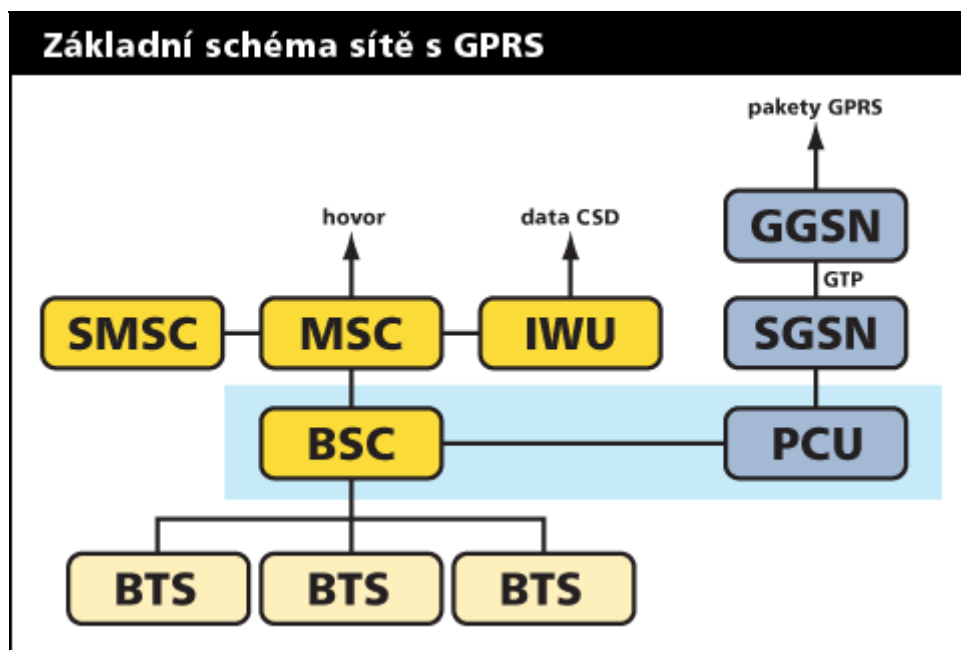
GPRS se hodí zejména pro využívání internetu díky své nesymetrické povaze, neboť při této činnosti tok dat v downstreamu (od serveru k uživateli) výrazně převyšuje data v upstreamu (směrem od uživatele k serveru). Výhodný je v tomto případě také mnohokrát diskutovaný způsob účtování služby. Pro čtení informací z internetu (prohlížení stránek) je vhodné účtování podle množství přenesených dat, na druhou stranu při jednorázovém přenosu může být někdy výhodnější účtování podle času. Technologie GPRS v principu dovoluje oba způsoby, popř. i jejich kombinaci. [13]

Pokud zhodnotíme podle obecných měřítek kvalitu služby GPRS (QoS – Quality of Services) a porovnáme ji s ostatními současnými mobilními technologiemi, vychází GPRS v našich podmínkách nejlépe. Bereme v úvahu aspekty jako je podpora technologie výrobců mobilních zařízení, podpora u operátorů, cena služby, dostupnost, přenosová rychlost, spolehlivost apod.

Tab. č. 2 – Přenosové rychlosti služeb GSM a GPRS

Služba	Up Link	Down Link
GPRS	13,4 kbps	28–64 kbps
GSM CSD	9,6–14 kbps	9,6–14 kbps
HSCSD	14,4 kbps	43,2 kbps

S podporou technologie GPRS u výrobců mobilů je to velmi dobré. Naprostá většina dnešních mobilních telefonů GPRS podporuje a neexistuje snad výrobce, který by se GPRS vyhýbal.



Obr. č. 1 – Základní schéma GPRS

Běžný přenos informace (hovor, SMS) v síti probíhá přes rádiové rozhraní mezi mobilem a základnovou stanicí BTS. Dále pokračuje přes řídicí část subsystému základnových stanic BSC do radiotelefonní ústředny MSC. Z ní může dále směřovat např. do veřejné telefonní sítě PSTN (Public Switched Telephone Network) nebo být předán do centra pro řízení krátkých textových zpráv SMSC (Short Message Service Center).

Při přenosu dat klasickým způsobem s přepínáním okruhů jsou data v ústředně předávána do jednotky označované zkratkou IWU (Inter-Working Unit). Obsahuje analogové nebo digitální modemy, které zprostředkují datovou komunikaci mezi GSM a jinými sítěmi.

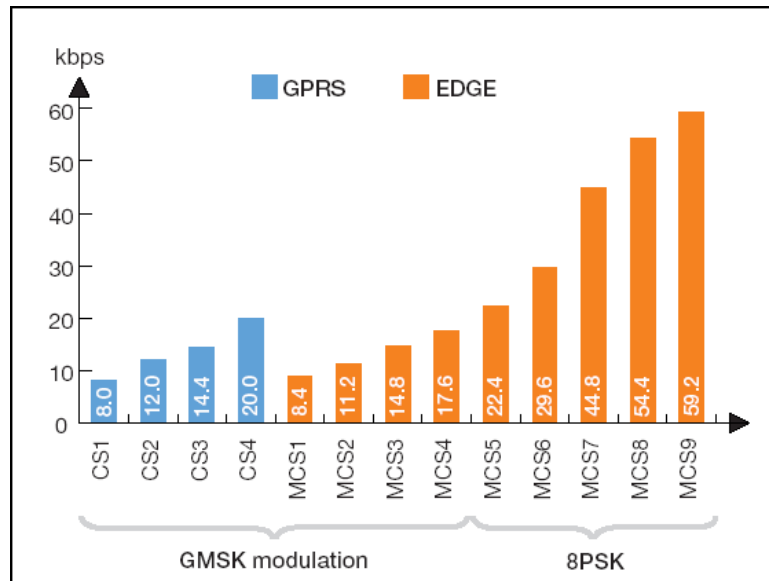
V případě GPRS, tedy paketového přenosu, probíhá komunikace samozřejmě opět mezi mobilním telefonem a BTS, ale v centru BSC jsou pakety předány do jednotky PCU

(Packet Control Unit), která se stará o identifikaci a řízení paketového způsobu komunikace na rádiovém rozhraní. Jednotka PCU je dále propojena s datovým uzlem SGSN (Serving GPRS Support Node). Datové uzly SGSN jsou schopny takřka přímo komunikovat s rádiovou částí sítě, tedy s řídicími jednotkami základnových stanic BSC (potažmo se základnovými stanicemi). Uzly SGSN mají za úkol např. zjistit polohu mobilního terminálu, ověřit jeho totožnost nebo se také starají o účtování za poskytnuté služby. Musí mít proto logicky přístup do některých systémových databází sítě. Např. do HLR (Home Location Register), což je v podstatě databáze účastníků sítě s řadou důležitých informací - mj. přístup ke službám operátora apod.

Uzel SGSN je dále napojen na další blok GGSN (Gateway GPRS Support Node), který je zde v úloze brány (směrovače paketů) a tedy komunikuje s jinými paketovými datovými sítěmi, např. s internetem (protokol IP). Přestože je na obrázku č. 1 pouze jedna dvojice SGSN a GGSN, síť může obsahovat, a reálně taky obsahuje, těchto dvojic více. Uzly mezi sebou komunikují prostřednictvím protokolu GTP (GPRS Tunneling Protocol), který pracuje nad síťovým protokolem IP. Toliko tedy o struktuře sítě. [14]

1.3 EDGE (Enhanced Data for GSM Evolution)

Je to technologie, která pouhou změnou modulace signálu dokáže několikanásobně zvýšit přenosovou rychlost stávající GSM sítě. Jen CSD připojení, které v GSM síti dosahuje přenosové rychlosti 9,6/14,4 kbps, má v síti EDGE přenosovou rychlost 48 kbps! Problém je ale v tom, že díky rozdílnému způsobu modulace je potřeba pro EDGE speciální telefon (respektive je nutné současný GSM telefon dovybavit demodulátorem kódování 8PSK, které EDGE používá). U EDGE existuje devět kódovacích schémat MCS1 až MCS9. Zatímco první čtyři se příliš neliší od schémat použitých u GPRS, u schémat MCS5 přichází ke slovu nová modulace a rychlosti rostou.



Obr. č. 2 – Porovnání modulací GPRS vs. EDGE

Výše uvedený graf porovnává kódovací schémata a jejich rychlosti u GPRS (modré sloupce) a EDGE (oranžové sloupce). U schémat MCS1 až MCS4 se používá stejná modulace jako v současných sítích (GMSK), u vyšších schémat je použita nová modulace 8PSK. Čísla udávají maximální rychlost na jeden timeslot, kterých lze najednou teoreticky využít až osm.

Vzhledem k tomu, že EDGE je opravdu založeno pouze a jen na změně modulace, tak jeho ostatní funkce a možnosti jsou naprosto shodné se standardními sítěmi GSM. V sítích EDGE tak najdete jak technologii HSCSD (v sítích EDGE se jí říká ECSD), tak technologii GPRS (v sítích EDGE se jmenuje EGPRS).

Tab. č. 3 – Porovnání GPRS vs. EDGE

Služba	GSM/GPRS	EDGE
CSD	9,6/14,4 kbps	48 kbps
HSCSD (2+2)	28,8 + 28,8 kbps	96 + 96 kbps
HSCSD (3+1)	14,4 + 43,2 kbps	144 + 48 kbps
GPRS	až 128 kbps	až 384 kbps

Z pohledu uživatele se s přechodem na EDGE až na zvýšenou rychlost téměř nic nezmění. Stále bude používat stejné internetové služby, nanejvýš přibudou další, které s „pomalým“ GPRS není možné využívat. [11]

Nejvíce se hovoří o možnosti přenosu videa, ale tato funkce je podle všeho jen marketingový tah než prakticky využitelná služba. Důležitější je ale například možnost přímého přístupu k Microsoft Exchange, což s GPRS příliš nedaří a rychleji se načítají webové stránky. Navíc také není problém v případě potřeby stáhnout soubor o velikosti několika MB.

EDGE ale není nutné využívat jen při spojení telefonu s počítačem. Přímě v mobilu jsou přece e-mailové a wapové prohlížeče, rychleji se odesílají multimediální zprávy. EDGE se také vcelku hodí pro novou službu Push to talk, kdy telefon funguje jako vysílačka.

1.4 Síť UMTS – mobilní síť třetí generace

Je to zkratka anglického Universal Mobile Telecommunication System. Označují se tak určité sítě třetí generace, které nabízejí oproti běžným mobilním sítím GSM zejména rychlejší datové přenosy a možnost videohovorů [32].

UMTS je mezinárodní standard vyžívající standardně frekvenční spektrum od 1885 MHz do 2025 MHz nebo od 2110 MHz do 2200 MHz. Jeden kanál má přesně definovanou šířku 5 MHz. UMTS se někdy také nazývá WCDMA (Wideband CDMA). Je to v základu totéž, ale toto označení zahrnuje i japonskou síť FOMA společnosti NTT DoCoMo, která s UMTS kompatibilní není. My se tedy přidržíme názvu UMTS. [33]

1.4.1 Dělení sítí UMTS

UMTS se dělí na dva základní typy, a to podle způsobu řešení duplexního provozu.

- Prvním, významně rozšířenějším typem je FDD (Frequency Division Duplex), kde je pro uplink a downlink použit odlišný kanál.
- Druhý je pak TDD (Time Division Duplex), kde uplink a downlink používá jeden kanál, ve kterém se oba směry střídají v čase. UMTS typy, které lze nalézt v současných telefonech na českém trhu a které nabízí samozřejmě již i řada zahraničních operátorů, jsou UMTS FDD.

1.4.2 Struktura sítě UMTS

Sítě UMTS mají velmi podobnou strukturu jako síť GSM. Základní stanice v UMTS se stále může nazývat BTS nebo BS (Base Station), nicméně obecně používané pojmenování je Node B. Každý Node B je pak připojen k RNC (Radio Network Controller), což je

obdoba BSC (Base Station Controller) v sítích GSM. Několik Node B a jedno RNC pak dohromady vytváří RNS (Radio Network System) neboli radiovou část sítě UMTS.

Zde ale narážíme na jeden zásadní rozdíl oproti sítím GSM – zatímco GSM sítě mají pokrývat obvykle celé území daného státu, tak sítě UMTS budou budovány formou malých ostrůvků, které budou pokrývat místa, kde je o UMTS zájem - tedy velká města. Tyto ostrůvky ale nebudou - alespoň zpočátku - mezi sebou propojeny a pokrytí UMTS tedy nebude spojitě.

Uživatel UMTS, který vyjede z oblasti obsluhované daným RNS bude moci dále využívat některé (popravdě, chybět bude asi jen videotelefonie) ze služeb operátora, a to v jeho GSM síti. Každý tento ostrůvek je tedy obsluhován jedním RNS. Všechny RNS dané UMTS sítě pak dohromady tvoří radiové rozhraní sítě UMTS – UTRAN (UMTS Terrestrial Radio Access Network). [33]

1.4.3 Výhody UMTS

Jednou z velkých výhod UMTS je možnost handoveru (tj. předání probíhajícího hovoru) mezi GSM/GPRS/EDGE a UMTS částí sítě. V ideálních podmínkách tak budete v místech pokrytých signálem UMTS připojeni k internetu přes UMTS a v místech, kde UMTS není, budete používat pomalejší GPRS.

UMTS dělá z hlediska běžného uživatele zajímavý také fakt, že v okamžiku, kdy si chcete zavolat a GSM síť je přetížená, může vás automaticky přemístit do UMTS, kam se vejde více hovorů. Problémy s odesíláním SMS a s voláním na Štědrý večer a Silvestra tak může UMTS poměrně elegantně vyřešit (za předpokladu, že v jádru sítě a na páteřních spojích je dostatečná kapacita).

1.4.4 Služby UMTS FDD

UMTS FDD má naprosto geniální vlastnosti, co se zacházení s hlasovými hovory týče. Má významně větší kapacitu než GSM, umí dynamicky měnit velikost buňky a nabízí i o něco lepší kvalitu hovoru. Nicméně kvůli tomu si UMTS pořizují operátoři jen zřídka (nicméně za zmínku stojí Velká Británie, kde byl velký problém s kapacitami GSM sítí a některé průmyslové oblasti Německa). Ostatně právě ve Velké Británii se některé ceny za hlasová volání v UMTS právě z těchto důvodů dostaly dokonce pod úroveň cen za volání v GSM.

Další službou, kterou v našich končinách nabízí pouze UMTS, je videotelefonie. Použitelnost této služby je poněkud sporná, protože při jejím využívání pochopitelně nemůžete držet telefon u ucha. To znamená, že zároveň potřebujete buď hlasité hands-free, které ale můžete použít jen v tichém prostředí nebo náhlavní soupravu, kterou zase obvykle moc lidí s sebou nenosí.

Poslední z nabízených služeb jsou samozřejmě datové paketové přenosy, které jsou, alespoň teoreticky, o něco rychlejší než v sítích GSM/GPRS/EDGE. Přenosová rychlost se ale u UMTS velmi podstatně liší v závislosti na rychlosti pohybu. I když zde je poměrně důležité rozlišovat UMTS R99 (tedy první UMTS standard) a jeho upravenou variantu UMTS R4. U obou ale platí premisa, že pomalu se pohybující objekty mohou využívat rychlost maximálně 384 kbps a rychle se pohybující objekty (nad 120 km/h) mohou dosahovat rychlosti maximálně 144 kbps.

1.5 Satelitní mobilní síť Iridium

Iridium je první společnost, která si vzala za své přiblížit satelitní telefonování koncovým zákazníkům. Vznikla v roce 1997 jako konsorcium několika velikánů v oblasti mobilní telekomunikace (např. Motorola, Sprint, Kyocera a další), které hodlali do společného projektu investovat něco kolem 5 miliard dolarů. Iridium předpokládala v prvním roce provozu cca 650 000 uživatelů s očekávaným nárůstem na pět milionů v roce 2002, přičemž se obecně předpokládalo, že celkový počet uživatelů satelitní komunikace dosáhne

v roce 2002 plných 12 milionů. [12]

Protože však satelitní síť Iridium měla ve svém nástupu řadu potíží, došlo k jejímu dočasnému útlumu. V roce 2001 však došlo k převzetí této sítě firmou Iridium Satellite LLC, která se rozhodla změnit koncepci celé sítě. Nová koncepce se zaměřila na nový okruh zákazníků, jako jsou například letecké přepravní firmy, které pomocí této sítě například navigují své letouny a dále je to armáda, která tuto satelitní síť využívá pro navigaci pozemních, vzdušných a námořních jednotek. [29]

2 POPIS EMBEDDED TECHNOLOGIE

2.1 Embedded technologie

Embedded technologie je ze své podstaty takový technologie, která je zabudovaná přímo do řízeného systémů, je jeho součástí. Už samotný název to napovídá, neboť v anglickém jazyce znamená „embedded“ „vestavěný“. Dva hlavní rozdíly mezi embedded a klasickým systémem jsou cena, odběr elektrické energie a výpočetní výkon.

Cena embedded systémů bývá obvykle vyšší, neboť tyto systémy musejí splňovat náročná kritéria na kvalitu provedení, stabilitu provozu, spolehlivost a povolenou zátěž. Protože jsou však v současné době embedded systémy vyráběny ve velikých sériích, dokáží se výrobci dostat při větších odběrech i na velmi příznivé ceny. Obecně však platí, že bývají dražší než standardní technologie. Odběr elektrické energie je druhé kritérium, které pomáhá specifikovat embedded systém. Spotřeba elektrické energie bývá u těchto zařízení obvykle velmi malá, řádově jednotky Wattů.

Výpočetní výkon bývá obvykle trochu pozadu za standardními platformami výpočetní techniky, a je to zejména z toho důvodu, že jsou na tuto technologii kladeny vysoké nároky na kvalitu a zátěž, což se při větších taktovacích frekvencích hůře dodržuje. Naštěstí to však není takový problém, jak by se mohlo na první pohled zdát. Úlohy, pro které bývají využívány embedded technologie totiž obvykle nevyžadují vysoký výpočetní výkon, spíše však vyžadují stabilitu a spolehlivost.

Výše uvedené vlastnosti embedded systému vedou jeho vývojáře programového vybavení ke specifickému přístupu k programování. Programy v embedded systémech často musejí běžet v reálném čase s limitovanými hardwarovými prostředky, často nemají žádný rotační pevný disk – tedy dostatečný permanentní úložný prostor, GUI (Graphical user interface) operační systém, natož tak klávesnici či monitor. Pevné disky bývají nahrazovány pomalejšími flash-pamětmi, klávesnice bývají nahrazeny obyčejnými jednoúčelovými key-pady a místo monitoru jsou často využívány LCD nebo TFT displeje.

Pro vývojáře jsou tedy tyto technologie v podstatě výzvou, neboť připravit fungující aplikaci pro embedded systém může být velmi komplikované.



Obr. č. 3 – TFT Panel PC Kit
- PPC 1211T

Jejich základní programová vybavení, jako bývá firmware či BIOS , se mohou nacházet jak na ROM tak na Flash pamětech či IC čípech. Programy často bývají psány ve strojovém jazyce (Assembleru) nebo ve vyšším programovacím jazyce, jako je ANSI C/C++, ovšem speciálně upraveného pro potřeby daného zařízení. Velikost programového vybavení, včetně výkonného programu, bývá často udávána v jednotkách stovek kB nebo jednotek MB. Jako příklad mohou složit Windows CE 2.11, které jako celek v minimální konfiguraci zabírají pouze 1,7 MB a přesto nabízí všechny ovladače pro základní rozhraní pro práci s daným zařízením, jako jsou paměť, generátor přerušení, sběrnice PC/104, Flash Disk, sériové a paralelní porty, grafický adaptér.

Protože embedded zařízení bývají velmi kvalitně vyrobena tak jejich pracovní podmínky mohou být mnohem horší než u běžných PC – jejich pracovní teploty se mohou pohybovat od -25 °C do +85 °C. Také jejich ukazatel poruchovosti, MTBF (Mean Time Between Failure – střední čas mezi poruchami), bývá velmi dobrý, dosti často až 50 000 hodin nepřetržitého provozu. Dá se říci, že ve většině případů se dají tyto zařízení řadit mezi IPC kompatibilní.

2.2 Historické kořeny embedded technologií – její začátky v 60. letech minulého století

Historické kořeny embedded technologií sahají do 60. let minulého století. V té době již byla známa koncepce integrovaných obvodů, které poprvé navrhl Jack Kilby z americké firmy Texas Instruments v roce 1958. Vznikla generace integrovaných obvodů generace SSI – „Small-Scale Integragion“, čili integrované obvody nízké úrovně integrace.

embedded systémy nebo systémy, které současnému chápání těchto systémů odpovídaly, se již tehdy částečně používaly v mainframe počítačích SYSTEM/360 z roku 1964 od firmy IBM. Tyto obsahovaly dávkový operační systém, který v sobě obsahoval částečně BOS (Basic Operating System), TOS (Tape Operating System) a částečně i DOS (Disk Operating System).

Největší nástup embedded systémů byl však až v době, kdy došlo k expanzi vesmírného programu NASA a tím pádem vznikla i velká potřeba vzniku malých, soběstačných a nízkoenergetických systémů, které by umožnily řídit vesmírné lety, zejména navádět vesmírnou raketu dle zadaných parametrů.



Obr. č. 4 – Nosná
raketa projektu Apollo 11

Jako první takovéto zařízení, které se dá považovat za průkopníka embedded systémů, je „Apollo Guidance Computer“, tj. naváděcí počítač Apollo, který sloužil k navádění vesmírných letů Apollo. Byl vestavěn přímo do korpusu rakety a tím pádem se mohl honosit přívlastkem „vestavěný“ – tedy „embedded“. Byl vyvinut Charlesem Starkem Drapperem v kooperaci s MIT Instrumentation Laboratory. Tento elektronický prvek obsahoval

důležité inerciální řídicí systémy. Každý let na Měsíc byl vybaven dvěma těmito zařízeními. Jedno zařízení mělo na starosti hlavní řídicí modul a druhé řídilo lunární modul.



Obr. č. 5 – Lunární modul
Apollo 11

Tento „počítač“ měl pracovní taktovací frekvenci 2,048 MHz, řídicí frekvence pro sběrnici byla 512 kHz, měl čtyři 16 bitové registry, dostupná pracovní paměť RAM byla 2 kB a paměť ROM, obsahující řídicí kód, byla 22 kB. Dále měl tento počítač k dispozici 5 vektorů přerušeni a šestřící „STAND-BY“ režim.

Když elektronika uspěla v programu Apollo, její vývoj dále pokračoval mílovými kroky. Všechny výsledky, jak je již zvykem nejen v USA, ale i obecně ve světě, pohltila armáda a dalším zařízením, které se může honosit zařazením do embedded technologií, byl naváděcí systém nukleárních raket Minuteman I a II. Tento systém byl ještě o něco výkonnější než „Apollo Guidance Computer“ – měl navíc pevný disk s 4 kB paměti.



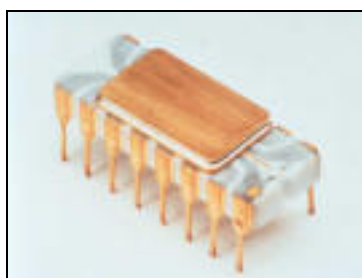
Obr. č. 6 – Odkrytý korpus
nukleární střely Minuteman
I s pohledem na řídicí elek-
troniku

2.2.1 Vývoj elektroniky a embedded technologií v 70. letech minulého století

Když během 70. let vrcholila studená válka mezi Sovětským Svazem a Spojenými státy americkými, embedded technologie byly považovány za státní tajemství a využívaly se zejména ve vesmírných programech NASA nebo vojenských programech Ministerstva obrany USA. Průnik těchto technologií do komerční sféry byl velmi pozvolný. A přestože elektronika dodávaná na trh z komerční sféry sahala vesmírným a vojenským technologiím sotva po kotníky, bylo na její vývoz uvaleno státní embargo a její prodej byl monitorován.

Na konci 70. let minulého století došlo k dalšímu pokroku ve vývoji integrovaných obvodů, a to vznikem technologie integrace střední úrovně – MSI (Medium-scale Integration). Tato technologie integrovaných obvodů již byla velmi zajímavá, protože i při vyšší ceně za jednotku dokázala snížit náklady na výrobu elektronických zařízení.

K dalšímu pokroku ve vývoji integrovaných obvodů a to vznikem technologie rozsáhlé úrovně integrace – LSI (Large Scale Integration). Vzápětí na to, v roce 1971, firma Intel přišla na trh se 4 bitovým mikroprocesorem Intel 4004. Stal se prvním mikroprocesorem na světě. Tento základní mikroprocesor měl 12 bitové adresování, 8 bitové instrukce a 4 bitové slovo. Jeho instrukční sada se sestávala z 46 instrukcí a taktovací frekvence měl 108 kHz. Cena tohoto mikroprocesoru byla již někde okolo 300 USD. Jako embedded zařízení sloužil v zařízení zvaném Busicom Calculator.



Obr. č. 7 – Intel 4004

Na jeho základě začaly brzy vznikat další, robustnější a výkonnější procesory, které jsou již obecně známější, neboť se staly na trhu s elektronickými součástkami velmi úspěšné. Jako prvním je nutno zmínit Intel 8008, který byl představen v roce 1972. Ten měl již taktovací frekvenci na 500 kHz, 8 bitovou sběrnici a mohl adresovat 16 kB. Ihned po svém

uvedení na trh se začal objevovat v jednoduchých terminálech, elektronických pokladnách a například v systémech na plnění lahví.

2.2.2 První 8 bitový mikroprocesor Intel 8080

V roce 1974 představil Intel čistě 8 bitový Intel 8080, který byl již opravdu plnohodnotný mikroprocesor. Pracoval s taktovací frekvencí 2 MHz, dokázal adresovat 64 kB paměti a mohl mít až 256 I/O zařízení. Tento mikroprocesor se objevil v počítačích MITS Altair 8800 a IMSAI 8080, které se staly zakladateli CP/M operačních systémů (Control Program/Monitor nebo -/Microcomputer). Tyto operační systémy vyvíjela firma Digital Research. V 70. letech se staly CP/M systémy na trhu dominantními. Jako řídicí jednotka embedded systému se Intel 8080 objevil například v kalkulačkách, řídicích systémech dopravních semaforů, ale kupříkladu se používal také k navádění raket.



Obr. č. 8 – Intel 8080A

2.2.3 Motorola HCF 6800

V roce 1975 přišla na trh firma Motorola s konkurenčním 8 bitovým procesorem Motorola 6800. Tento mikroprocesor měl novou, do té doby nepoužitou, vlastnost, a to Index Register a sběrníkovou technologii HCF (Halt and Catch Fire). Tento mikroprocesor se stal základem mikropočítače SWTPC 6800 (SouthWest Technical Products Corporation). Tento mikropočítač se stal brzy dostupný široké veřejnosti. Firma nadále pokračovala ve vývoji mikropočítačů a stala se výrobcem počítačových terminálů, počítačových beden, procesorových karet, paměťových karet, základních desek, I/O karet, diskových a páskových systémů. Vývojová řada mikroprocesorů Motorola 6800 byla nadále vyvíjena a postupně se stala součástí herních video konzolí Vectrex a mezi ostatními také součástí počítače TRS-80 Color Computer.



Obr. č. 9 – Motorola
XC6800A

V roce 1976 vznikl další velmi populární 8 bitový mikroprocesor. Vyrobita jej firma Zilog a jmenoval se Z80. Stal se součástí velkého počtu osobních mikropočítačů a herních konzolí. V době své slávy byl známý především tím, že byl nasazen v mikropočítačích Sinclair ZX Spektrum. Mutace tohoto mikropočítače se objevily také ve státech bývalého východního bloku jako Didaktik Gama a Didaktik Delta.

2.2.4 První víceúčelový mikrokontrolér Intel 8051

V roce 1980 firma Intel vyrobila 8 bitový mikroprocesor Intel 8051, který se záhy stal velmi populární součástí nově vzniklých embedded systémů, a to proto, že kromě nízké ceny se může pochlubit tím, že má sám o sobě na svém čipu implementovanu řadu periferních zařízení, jako jsou počítadla, časovače, UART paměť a samozřejmě také 4 kB vestavěné programové paměti a 128 B vestavěné datové paměti. Může adresovat až 64 kB paměti mimo čip.



Obr. č. 10 – Původní Intel 8051 ve
40 pinovém pouzdře

Je velmi vhodný pro velké množství řešení technologických problémů a počítá se s tím, že jedna polovina všech dodnes existujících embedded zařízení pracuje právě s tímto mikroprocesorem nebo s jeho náhradou. Jeho upravené verze jsou dodnes vyráběny firmami Dallas Semiconductor, Atmel, Philips a mnoha dalšími.



Obr. č. 11 – JS2051 –
Intel 8051 kompatibilní

V roce 1986 byl poprvé vytvořen paměťový modul RAM o velikosti 1 MB a hned v roce 1987 byl poprvé představen mikroprocesor Intel 80386. Jeho verze DX byla plně 32 bitová, verze SX pouze vnitřně, externě byla pouze 16 bitová a konečně tento mikroprocesor existoval i ve verzi SL – což byla verze pro přenosné počítače, která měla sice menší výkon, ale hlavně menší spotřebu elektrické energie. Intel 80386 představoval již třetí generaci procesorů Intel. Stal se na dlouhou dobu standardem, protože již měl implementovanou mohutnou instrukční sadu a měl možnost adresovat až 4 GB operační paměti, virtuální paměť v chráněném režimu mohla mít až 64 TB. Taktovací frekvence byla od 16 do 33 MHz a měl 275 tisíc tranzistorů, každý o velikosti 1 μm . Podle údajů firmy Intel mohl tento procesor vyhledávat klíčová slova v Encyklopedii Britannica během 12,5 sec.

2.2.5 Počátek věku informačních technologií v 90. letech minulého století, nástup miniaturizace a hromadné využívání elektroniky v průmyslu

V devadesátých letech minulého století, možná díky ukončení studené války, nastal velký zlom ve vývoji výpočetní techniky. Nastala doba rychlého vývoje technologií výpočetní techniky. Začalo se hovořit o nástupu doby informačních technologií. Nyní se již intervaly mezi uvedením nových technologií tak zkrátily, že výpočetní techniky a elektronika vůbec začaly zastarávat již několik měsíců po uvedení na trh.

Jako prvním známkou pokroku bylo vytvoření standardu sběrnice „Peripheral Component Interconnect“ známého spíše pod zkratkou PCI, a to v roce 1990 firmou Intel. Ovšem první komerční verze PCI 1.0 byla dodána na trh až v roce 1992. Tato sběrnice se stala novým standardem systémových sběrnic a její vylepšené varianty se používají dodnes. Poslední z řady PCI sběrnic je tzv. „Hot-Plug PCI“ nebo také „PCI-X“, která umožňuje vyjmát a vkládat zařízení přímo za chodu počítače.

V roce 1992 také vznikla již v pořadí třetí verze sběrnice SCSI a to SCSI-3. Tato již byla 32 bitová a dosahovala přenosových rychlostí až do 40 MB/s.

Když firma Intel dodala na trh svůj první mikroprocesor Intel 80486, bylo již zřejmé, že se dlouho tento výrobek na výslunní slávy neohřeje a že bude brzy nahrazen technologií o řád lepší. Začal závod o překonání hranice taktovací frekvence 100 MHz. Tato byla pokořena v březnu roku 1993 opět firmou Intel, která dodala na trh mikroprocesor Intel 80586, který byl přímým nástupcem Intel 80486 a dostal označení Pentium. Vzápětí v roce 1995 dodala na trh firma AMD svůj první výkonný mikroprocesor AMD 5x86. Tento byl sice postaven na architektuře Intel 80486, ale dokázal pracovat na taktovací frekvenci 133 MHz a vyrovnal se Intel Pentiu 75 MHz. Podobně v témže roce dodala firma Cyrix na trh svůj Cyrix 5x86, který dosahoval frekvencí 100-166 MHz, ale míval problémy s teplotou a ne každý jednotlivý kus byl stabilní pro výrobcem stanovenou taktovací frekvenci.



Obr. č. 12 – AMD
5x86

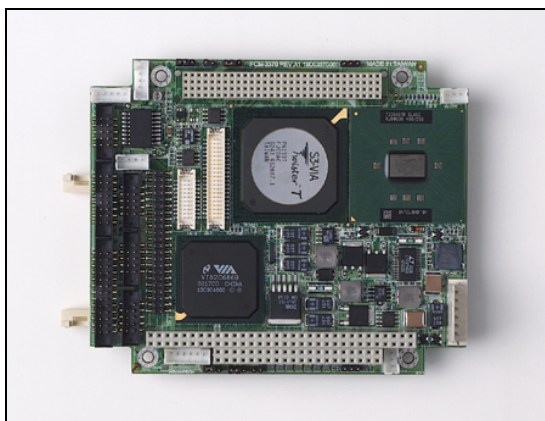
Tím, že na trhu s mikroprocesory začaly existovat tři velké dominantní firmy, začala doba rychlého vývoje a stlačování cen nových mikroprocesorů na minimální mez. Tento trend začaly kopírovat i další odvětví závislé na informačním průmyslu a od tohoto roku v průměru každý půlrok vznikl jeden nový typ mikroprocesoru od jednoho, dvou nebo i všech tří dominantních výrobců. Samozřejmě že k novým mikroprocesorům vznikaly dříve či později i vhodné základní desky, dostatečně rychlé RAM paměti a další komponenty.

2.2.6 Vznik sběrnice PC/104

Důležitým mezníkem pro moderní embedded systémy bylo v únoru roku 1992 vytvoření konsorcia PC/104, které definovalo a zavedlo standard sběrnice PC/104. Tento standard byl zaveden a schválen 12 firmami jako trend, kterým se bude ubírat vývoj moderního embedded zařízení. V současnosti je již pod standardem PC/104 podepsáno přes 100 firem, a toto číslo neustále narůstá.

Co to však PC/104 je? Jedná se o otevřený design sběrnice nabízející výkon a flexibilitu IBM kompatibilního počítače ve velikosti, která ideálně vyhovuje embedded systémům. Původní návrh sběrnice PC/104 počítal s kompatibilitou se stávající sběrnici ISA a EISA (Enhanced Industry Standard Architecture). Ovšem tento standard byl brzy překonán a kvůli požadavku kompatibility se sběrnici PCI vznikl v roce 1997 vylepšený standard PC/104-Plus. Tento již v současné době vyhovuje velké většině požadavků, které na systémovou sběrnici kladou současné embedded systémy.

Na obrázku č. 13 je zobrazena PC/104 deska PCM-3370 od firmy Advantech® s mikroprocesorem embedded Intel Pentium III s taktovací frekvencí 800 MHz, instalovaným systémem Watch-Dog, vestavěnou AGP 4x SVGA kartou, podporou sériových portů, ethernet adaptérem a samozřejmě s podporou PC/104 a PC/104 Plus. Více o tomto zařízení se nachází v příloze P IIIA a B a popis sběrnice PC104 v příloze P IVA a B. [6]



Obr. č. 13 – Advantech PC/104 Plus
CPU module PCM 3370

2.2.7 Universal Serial Bus - USB

V roce 1995 byl vyvinut standard „Universal Serial Bus“ – USB, který vznikl jako nástroj umožňující nahradit stávající IDE a PCI sběrnice. Umožňuje připojovat a odpojovat zařízení k/od počítače za jeho chodu (Plug & Play), obsahuje též možnost je napájet a tím je učinit nezávislé od externího zdroje energie. USB 1.0 nabízela přenosové rychlosti až 12.5 Mbps a brzy na to vzniklá USB 2.0, která byla uvedena na trh v roce 2000, umožnila využít přenosovou rychlost až 480 Mbps. K tomuto druhu sběrnice se dají připojit různé druhy externích zařízení, od myši, klávesnic, přes WEB kamery, scannery, tiskárny až po

velkokapacitní datová zařízení, jako jsou hard-disky a CD-ROM, CD-RW a DVD-ROM mechaniky.

2.2.8 Současná výpočetní technika

V současné době, tj. v roce 2006, dominují trhu s mikroprocesory firmy Intel a AMD, které dodávají na trh mikroprocesory s taktovacími frekvencemi jádra běžně nad 1,5 GHz, a standardními mikroprocesory se staly řady AMD Athlon 64 a Intel Core 2 DUO.

2.3 Architektura embedded systémů

Procesor je napojen na svou sběrnici, na které se nachází paměť cache a paměti RAM (Random access memory) a ROM (Read Only Memory) a řadič přerušení. Na tuto sběrnici je napojen řadič systémové sběrnice, která může být v případě embedded systémů typu ISA, PC/104 nebo PC/104-Plus.

Na systémovou sběrnici se potom dále napojují jednotlivá I/O zařízení, jako je CMOS (Complementary Metal Oxide Semiconductor) paměť, sériové a paralelní porty, port PS/2, řadič pevného disku nebo flash disku, řadič ethernet adaptéru, videa a také třeba zvuku. Na systémové sběrnici mohou být dále dodatečná zařízení vyrobená speciálně pro daný embedded systém, která rozšiřují jeho schopnosti dle potřeb. Rozsah připojených externích zařízení je opět závislý na konkrétních potřebách pro daný systém.

2.3.1 Dostupné platformy mikroprocesorů pro embedded systémy

Klíčový prvek embedded systému je mikroprocesor. I když se platforma vybraného mikroprocesoru u každého jednotlivého výrobce a typu zařízení liší, v současnosti existují zejména tyto základní platformy procesorů pro embedded systémy:

a) x86

Platforma založená na rodině mikroprocesorů řady Intel x86. Je všeobecně často užívaná pro průmyslová embedded PC a mikrokontroléry Intel. Mikroprocesory řady x86 v embedded zařízeních bývají často upravovány tak, aby měly co nejvyšší poměr výkonu ku odběru elektrické energie. V současnosti se mezi nejsilnější mikroprocesory této platformy řadí embedded Intel Pentium IV. [17]

b) ARM, Strong ARM

„Advanced RISC Machine“ - Platforma založená na 32 bitové technologii, která se velmi často objevuje v kapesních počítačích a elektronických diářích. Je známá tím, že se může každá jeho instrukce stát podmíněnou. Z toho plyne, že strojový kód programů pro tuto platformu bývá kratší než u jiných platform. Platforma Strong ARM je modernizovaná verze, která se objevuje jak ve 32, tak i v 64 bitové verzi. [18]

c) PPC

„Power PC“ - platforma vyvinutá v roce 1991 konsorciem Apple – IBM – Motorola, známým též pod zkratkou AIM. Jedná se o RISC architekturu procesoru inspirovanou architekturou MIPS. Mikročipy této vývojové řady existují jak ve 32 tak i ve 64 bitové verzi. Firma IBM dodává embedded systémy s touto architekturou v provedení „system-on-chip“, a to řadu 4xx – z ní nejnovější je model 440 GX.

Firma Motorola dodává na trh série MPC 5xx, 8xx, 82xx a 85xx. Nejvýkonnější z těchto sérií je MPC 8560, vybavená mikrokontrolérem Power QUICC III a s vestavěnou podporou komunikačních a síťových prvků. [19]

d) HPC

„Handheld PC“ - platforma 32 bitových procesorů vyvinutých firmami Compaq a Hewlett & Packard. Používá se zejména u kapesních diářů a kapesních počítačů.

e) MIPS

„Microprocessor without Interlocked Pipeline Stages“ - platforma firmy MIPS technologies. Jedná se opět o 32 bitovou RISC architekturu mikroprocesoru. Byla vyvinuta v roce 1985 a nasazena poprvé do mikroprocesoru R2000. Existuje v řadách MIPS I, II, III, IV, 32 a 64. Jeden z posledních mikroprocesorů této řady byl představen v roce 1995 – a byl to R16000 – měl až 8 MB vnitřní cache a taktovací frekvence do 700 MHz. Používá se u různých přenosných embedded zařízení – mohou to být kapesní diáře, kapesní počítače, herní konzole (Nintendo, Sony PlayStation) a podobně. [20]

f) SH1 až SH4

„System Hitachi“ – SH1 a SH2 jsou platformy 16 bitových a SH3 a SH4 jsou platformy 32 bitových RISC mikroprocesorů firmy Hitachi. Výhoda těchto procesorů je nízká spotřeba elektrické energie – řádově v μW , proto je tato platforma běžně užívaná

v miniaturních kapesních diářích, miniaturních kapesních počítačích a mobilních telefonech. [21]

g) SPARC

„Scalable Processor ARChitecture“ – RISC architektura mikroprocesorů vyvinutá v roce 1985 firmou Sun Microsystems. SPARC je registrovaná obchodní značka firmy SPARC Inc., která vznikla v roce 1989, aby podpořila tuto technologii a umožnila její kvalitní testování a úspěšné nasazení na trh. Tato technologie byla například využita v licenci známými firmami jako jsou - Texas Instruments, Cypress Semiconductor a Fujitsu. [22]

2.3.2 RISC architektura

„Reduced Instruction Set Computer“ – počítač s redukovanou instrukční sadou. Architektura mikročipů vyvinutá v 70. letech minulého století. Odborníci na vývoj mikroprocesorů v té době budovali své nové procesory příliš složité se zbytečně velkou instrukční sadou, z níž velká většina programů vytvořených pomocí tehdejších kompilátorů využívala jen malou část. Proto tehdejší vývojáři mikroprocesorů věnovali svůj čas na ladění nejvíce používaných instrukcí a stávalo se, že hromadné instrukce pracující s více operandy bývaly často pomalejší než sekvence instrukcí menších.

Na tuto skutečnost poukázal profesor Andrew Tanenbaum, profesor na univerzitě Vrije v Amsterdamu a navíc ještě dokázal, že na 98 % konstant v kódu tehdejších aplikací stačilo pouhých 13 bitů. Dal tedy impuls k tomu, aby inženýři vyvinuli novou architekturu mikroprocesorů, které by měly menší instrukční sadu, ale dokázaly v jednom taktu zpracovat více operandů a více výpočetních a logických operací. Dospělo se tedy ke vzniku technologie redukované instrukční sady. V op-kódu každé instrukce tedy bylo více místa na operandy

a došlo tedy ke smrštění a zrychlení kódu. Ačkoliv první RISC mikroprocesory nebyly z nejrychlejších, následným vývojem a laděním došly do stádia, kdy se staly jedny z nejvýkonnějších na trhu a staly se základnou mnoha serverových stanic.

Níže je uveden pro tuto platformu vždy v prvním sloupci ukázkový kód v assembleru (registry jsou uvedeny jako indexy s dolarem), ve druhém sloupci adekvátní kód v programovacím jazyce C++ a v závorce je uveden význam:

- `add $1,$2,$3` `int i1 = i2 + i3;` (přičítání se znaménkem)

- `addu $1,$2,$3` `uint u1 = u2 + u3;` (přičítání bez znaménka)
- `sub $1,$2,$3` `int i1 = i2 - i3;` (odečítání se znaménkem)
- `subu $1,$2,$3` `uint u1 = u2 - u3;` (odečítání bez znaménka)
- `addi $1,$2,100` `int i1 = i2 + 100;` (přičtení konstanty k levému operandu a výsledek je přiřazen pravému)

Ohromná výhoda této architektury je v tom, že mikroprocesor dokáže v jednom taktu udělat to, co obyčejný procesor dělá minimálně takty dva. Je tedy velmi vhodný pro embedded aplikací, neboť u nich hraje velkou roli velikost programového a řídicího kódu. [23]

2.3.3 Operační systémy, vývojová prostředí

V současnosti jsou k dispozici čtyři základní druhy operačních systémů, které se dají využít pro tato zařízení, a to:

- UNIX

Jeho nevýhodou je, že jeho samotná inicializace vyžaduje maximální odbornost a tvorba programů pro tento druh operačního systému je zdlouhavá a vyžaduje velké zkušenosti. Další nevýhodou je, že operační systém UNIX je pod ochranou autorských práv a tím pádem se musí pro jednotlivé CPU platformy kupovat licence.

Výhodou naopak je, že tvorba programů pro tento operační systém je již známa delší dobu a existuje hodně zdrojů, odkud čerpat pomocné informace. [24]

- Linux

Tento operační systém je na seznamu software umožňujícího řídit běh embedded zařízení teprve krátkou dobu. Proto jsou v současnosti možnosti tvorby konfigurovatelné systémové platformy a vývoje programů pro tento operační systém omezené.



Obr. č. 14 – Logo-
Embedded Linux
Consortium

Linux jako operační systém k embedded systémům je k dispozici ke stažení z internetu zdarma. [25]

- Microsoft Windows CE, Microsoft Windows CE .NET



Obr. č. 15 – Logo Microsoft ®
Windows Embedded

Operační systém Microsoft Windows CE a CE. NET je momentálně jediný na trhu, který umožňuje pohodlné vytváření konfigurací pro jednotlivé CPU platformy. V současnosti existují graficky orientované vývojové nástroje pro výrobu programového vybavení.

Výhodou je, že existují již předpřipravené systémové platformy pro jednotlivé CPU platformy, které se pouze nainstalují a vcelku pohodlným způsobem je možno měnit konfiguraci a komponenty v podstatě nově vzniklého operačního systému.

Nevýhodou je, že se za pořízení software potřebného k vývoji systémové platformy a programového zařízení musí platit licenční poplatky. Ukázka vývojového prostředí Microsoft Windows CE Tools 3.0 je v příloze PII.

- Symbian OS



Obr. č. 16 – Logo Symbian OS

Tento operační systém je dodáván s mobilními telefony a je přímo připraven pro činnost v těchto zařízeních. Momentálně tento operační systém podporují velcí výrobci mobilních telefonů Nokia, Motorola, Siemens, Sony-Ericsson a připravují se další, jako Panasonic a Samsung.

Výhodou je, že k vývoji programového vybavení existuje graficky orientované vývojové prostředí, které se dá pořídit přímo pro daný typ mobilního telefonu.

Nevýhodou je, že se opět za používání vývojového software musí zaplatit licenční poplatky. [26]

2.3.4 Ladění chodu embedded aplikací (debugging)

Debugging, čili ladění aplikací, je obvykle prováděno na softwarovém emulátoru embedded systému pomocí ladících služeb jádra embedded systému nebo v ladícím programu, který dokáže zasílat mikrokontroléru embedded zařízení žádosti o přerušování a tím vstoupit do jeho interního mikrokódu. Přerušování mikrokódu v embedded zařízení je tedy možné pouze na hardware obsahující žádanou CPU platformu nebo v emulátoru, který dokáže emulovat instrukce žádané CPU platformy.

Dále je možno přímo přerušovat činnost CPU pomocí přerušování mikrokódu přímo na zařízení. Jedná se o takzvané „na procesor zaměřené ladění“ (CPU-based debugging). Tento způsob umožňuje sledovat činnost mikrokontroléru na reálném hardware v reálném čase. Tato technologie ladění byla poprvé použita u zařízení PDP-11.

Vývojáři široce využívaná metoda je ladění programů přímo ve vyšším programovacím jazyku, kde je možno využít tzv. breakpointů a krokování po řádcích. Tyto možnosti již umožňují skoro všechna vývojová prostředí pro embedded systémy.

2.3.5 Postup a pravidla při psaní programů pro embedded systémy

Při psaní aplikací pro embedded systémy se všeobecně doporučuje tento postup a dodržování níže uvedených pravidel:

- Podrobné seznámení s architekturou embedded systému,
- vyhledání kvalitního vývojového prostředí (nutno brát zřetel zejména na komfort a vestavěné vlastnosti – jako způsoby ladění, překladu apod.),
- vytvoření vývojových diagramů, stanovení postupů prací a vytvoření analýzy problematiky,
- řádné komentování psaného kódu,
- před testováním programu na reálném embedded systému se doporučuje otestovat chod a funkčnost na emulátoru,

- při ladění na reálném embedded zařízení je nutno dbát zvýšené opatrnosti při práci s I/O zařízeními, některé operace s I/O zařízeními mohou mít fatální následky jak pro chod programu, tak i pro celý systém,
- ve finále je dobré nechat program pracovat samostatně bez zásahu programátora a až po určité stanové době bezchybné práce je možno tento program označit za korektní. Všeobecně se doporučuje testovat po dobu deseti až statisícinásobek délky doby jednoho pracovního cyklu programu.

2.3.6 Vytvoření systémové platformy, startovací rutiny a chod embedded systémů

Obecný embedded systém mívá své jádro kompilované do binárního souboru. Tento bývá často ještě komprimovaný, aby na cílovém zařízení zabíral co nejméně místa. Této kompilaci se říká image nebo foot-print systému. Image se obvykle nahrává do ROM paměti u jednocelových embedded systémů, u víceúčelových systémů se využívá EEPROM či Flash-disk.

V ROM paměti embedded systémů bývá obvykle i nahráno pravidlo zavádění systému v činnost. Tomuto se říká boot-strap leader. Obvykle to bývá binární kód mikroprocesoru, který vyvolá kód, který se nachází v paměti ROM na adrese 0x000000 a z tohoto místa se již provádí samotné zavádění systému.

Při zavádění systému se provede nejprve kontrola CRC (kontrolního součtu) image systému a pokud je vše v pořádku, začne se systém nahrávat z ROM či Flash disku do RAM. U inteligentních systémů se nahrají jen ty nejnnutnější soubory a ostatní se zavádějí až tehdy, kdy jsou potřeba. Jinak se provede dekomprese celé Image do paměti RAM. Image u standardních průmyslových embedded systémů bývá obvykle 500 kB až 2 MB velká. Ovšem například při použití konfigurace MAX-ALL u Windows CE 3.0 je možno dosáhnout i velikosti 12 MB.

U image velikosti 500 kB dojde k zabránění RAM o velikost zhruba dvou až čtyřnásobku její velikosti, tj. zhruba 2-3 MB, proto je nutno při kompilaci jádra systému dbát na výslednou velikost image a používat pouze komponenty, které jsou nezbytně nutné pro chod systému.

Po zavedení systému se provede inicializace samotného operačního systému a tento již bude mít k dispozici zbylou paměť RAM a všechna periferní zařízení, připojená k systému

v době inicializace a i k některým zařízením připojeným i po inicializaci. Nyní je embedded systém aktivní, má zaveden operační systém a může začít pracovat.

Samotný chod embedded zařízení je od této chvíle závislý na programech, které jsou spuštěny po startu operačního systému. Při správně napsaném programu a vhodně zvolené konfiguraci jádra systému je možno mít embedded zařízení v činnosti nepřetržitě například po dobu jednoho roku, aniž by toto zaznamenalo jakýkoliv hardwarový problém. [28]

2.3.7 Typy architektur embedded systémů

V současnosti existují tyto základní typy architektur embedded systémů:

a) Operační systémy s kontrolním cyklem (Control Loop OS)

V tomto návrhu operačního systému je software proveden tak, že má pouze jeden základní cyklus. Teprve z tohoto cyklu jsou volány podprogramy/subrutiny, které již samy o sobě spravují či řídí dané součásti hardware nebo software/operačního systému. Přerušení (Interrupt) pouze nastavují globálně dostupné příznaky, které jsou pak přístupné zbytku programu (nebo taky všem dalším programům v paměti)

Jednoduché API (Application Programming Interface) povoluje nebo zakazuje dané vektory přerušení. Správně použité přerušení pak volá vnořené subrutiny, které již obsluhují systémová volání a události, hardware a výjimky. Je to jedna z nejjednodušších metod vytvoření exokernelu.

V tomto typu operačního systému vždy existují třída podprogramů nebo subrutin, které obsluhují softwarově definované timery (počítadla, časovače) pomocí přerušení hodin reálného času. Tyto hodiny pracují s takovým rozlišením, jaká je taktovací frekvence procesoru nebo taktovací frekvence obvodu, který generuje reálný čas. Když daný timer doběhne, je zavolána subrutina spojená s tímto timerem nebo se nastaví daný příznak.

Očekávané hardwarové události by měly být zálohovány a připraveny k dalšímu zpracování právě s použitím softwarového timeru. Podle zkušeností může hardware selhat jednou z trilionu operací s ním provedených, což je zhruba asi jednou do roka. Přesto vynechání obsluhy hardware pomocí softwarového timeru může být časem velký problém.

Dále je důležité sledovat stav celého zařízení, aby bylo možno reagovat na nastalé situace co nejrychleji. Mnoho návrhářů/vývojářů těchto operačních systémů doporučují jedno čte-

ní z I/O zařízení v jednom průběhu cyklu s tím, že se daná hodnota uloží, a logika programu může na tento stav reagovat i později.

Je tu však první velká slabost, kterou tyto systémy trpí, a to že není možno zaručit čas, za který zařízení odpoví na danou událost. Opatrné vytváření kódu může jednoduše zajistit, že žádná událost nezdrží přerušeni na příliš dlouhou dobu. Tím pádem přerušeni může pracovat s velkou předsostí.

Další velkou slabostí je, že nelze jen tak jednoduše do tohoto systému přidávat další součásti. Algoritmy (postupy zpracování dat), které zabírají mnoho času, se musejí často rozdělit na více fází, z nichž každá jedna fáze běží v jednom průchodu cyklem. A pokud se daný algoritmus příliš zdrží, musí zde být šance jej slušně ukončit.

Ale naopak velká síla těchto systémů je v jejich jednoduchosti a nízkém nároku na paměť. Proto nikoho příliš nepálí, že při nešetrném zacházení s programem může být chování těchto systémů nepředvídatelné.

b) Operační systémy s multitaskingem bez nuceného přerušeni cyklu (Non-preemptive multitasking OS)

Tento druh operačního systému je velmi podobný jako výše uvedený operační systém založený na jednom hlavním cyklu, kromě toho, že tento cyklus je před programátorem aplikací skrytý v jádru systému (system core nebo kernel core) API. Tento cyklus je pro všechny aplikace běžící v tomto systému shodný, ale každý program, běžící v tomto operačním systému má vlastní stack (odkládací paměť pro proměnné a adresy funkcí). Dále má každý program (proces) příznak, který říká, v jakém stavu se daný program nachází a když je program neaktivní (idle), tak se volá funkce nebo podprogram, který obsluhuje tento stav (obvykle se nazývá „pause“, „wait“, „yield“ nebo podobně). Tento podprogram pouze kontroluje stav systému a když má být aktivován, prostě přejde z čekací smyčky zpět do svého výkonného kódu.

Dále v tomto operačním systému existují fronty zpráv/událostí, které jsou postupně zpracovávány a které, když nastanou, tak je procesu přiřazena priorita, dané zprávy jsou zpracovány a po dokončení zpracování je každá daná zpráva/událost smazána z fronty.

Výhody a nevýhody jsou velmi podobné jako u předchozího typu operačního systému, jen je tam velký rozdíl v tom, že přidávání nového software je mnohem jednodušší. Prostě se

napiše jádro nového programu a k ní se přidá nová fronta událostí a její obslužný program. [27]

c) Operační systémy s multitaskingem s nuceným přerušáním timeru (Preemptive timers OS)

Tento druh operačních systémů překonává všechny výše uvedené druhy operačních systémů, ovšem přidává navíc ještě timer (počítadlo) ke každému novému programu nebo procesu. Toto dává tomuto systému naprosto nové vlastnosti a možnosti. Až nyní je možno zaručit, že události budou zpracovány v garantovaném čase.

Navíc je tu ještě jedna novinka, a to, že program může vstoupit do svých datových struktur (vlastní paměti) v neočekávanou chvíli, a tedy pracuje právě tehdy, kdy je to třeba. Ovládací programy pro timer (časovač) jednotlivých programů musejí být ovšem vytvářeny se stejnou pečlivostí jako samotné ošetření rutin přerušení (interrupt). [27]

d) Operační systémy s multitaskingem s nuceným přerušáním úlohy (Preemptive tasks OS)

Tento druh operačních systémů překonává všechny druhy, které neobsahují nucené přerušování ani přerušovací timer. Je však trochu odlišný. Každý kousek kódu jedné úlohy může svým chodem narušit chod úlohy druhé nebo i celou jinou úlohu, musí se proto velmi dbát na oddělení jednotlivých úloh (programů).

Přístup na sdílená data musí být velmi pečlivě kontrolovaný pomocí nějaké synchronizační strategie, například pomocí semaforů nebo front zpráv. Místo tohoto typu operačních systémů je lepší pořídit si real-time operační systém, ale pokud nejsou dostupné zdroje pro pořízení real-time operačního systému, je tento typ operačního systému dobrá volba. [27]

e) Operační systémy kancelářského stylu (Office-style OS)

Tyto jsou populární pro embedded projekty, které nemají dostatečný rozpočet na systém. Většinou je však volba těchto druhů operačních systémů omyl, protože operační systémy tohoto druhu jsou speciálně zabalené knihovny obsahující použitelný kód. Pokud jsou však špatně napsané a obsahují chyby, pak je celý systém nepoužitelný.

Většinou tyto systémy postrádají interface, který by umožnil jednoduše vestavět podporu pro jiný než podporovaný hardware. Například pokud se využije operační systém Linux jako kontrolér servomotoru nebo telefonní ústředny, většina reálných kontrolních operací

skončí v jádru systému jako seznam očíslovaných funkcí s IOCTL voláním (Input/Output Control), a tak samotný operační systém brání vývoji nových aplikací. V podstatě je to proto, že operační systém „kancelářského-stylu“ chrání hardware od uživatelských programů.

Protože však většina embedded operačních systémů neprovádí „kancelářskou činnost“, většina kódu „kancelářských“ operačních systémů je k ničemu. Na příklad GUI rozhraní a podpora souborového systému je v těchto případech pouhým plýtváním místem.

No a protože se tyto operační systémy nasazují na reálný embedded hardware, musejí obsahovat ovladače pro všechna využitá hardwarová zařízení. Je tedy úplně jasné, že ani tady vývojář neušetří, protože musí do jádra „kancelářského“ operačního systému ovladače pro hardware dopsat nebo sehnat.

f) Exotické uživatelské operační systémy (Exotic OS)

Protože některé systémy vyžadují bezpečné a časově účelné chování, které se nedá docílit kombinací výše uvedených systémů, existují způsoby jak si navrhnout architekturu operačního systému vlastními silami. Pro to, aby bylo možno vytvořit si v podstatě novou platformu, je nutno najmout odborníka na platformu, na které se bude systém stavět, ušetří to měsíce ladění a přes jeho vyšší platové nároky budou ve finále náklady na vývoj nižší.

Aby se zjistilo, zda-li mohou zvolené úlohy pracovat nad definovaným hardwarovým řešením, dá se využít analýzy RMA („Rate Monotonic Analysis“). Tato ve své nejjednodušší formě předpokládá vytvoření testů, které ověří, zda-li z naplánovaných úloh ty nejrychlejší mají nejvyšší prioritu a mikroprocesor se využívá v průměru maximálně na 70 %. Harmonizace jednotlivých úkolů dokáže optimalizovat využití CPU. V podstatě se testuje, zda-li všechny programy v systému běží podle hlavního timeru. Ovšem je těžké toto docílit, protože většina real-time operačních systémů většinou přepínají úlohy ve chvíli, kdy se čeká na odezvu od I/O zařízení.

Dále se zjistí, zda-li existují v systému jen dvě úrovně priority – a to „běžící“ a „přerušeno nepovoleno“. Z principu nelze použít více úrovní priorit a ani semaforů.

Důležité je si uvědomit, že monitorovací systémy nesmějí mít nikde uvnitř software mrtvé body. Vždy musí existovat softwarové řešení všech stavů, které mohou nastat. Monitorovací rutiny musejí být co nejmenší a co nejrychlejší, aby celý systém běžel správně.

Takže v principu to znamená, že pokud systém využívá duální prioritu a monitorovací rutiny jsou bezpečné a důvěryhodné (neobsahují nikde mrtvé body), pak tento systém bude stabilní a bude pracovat dobře. [27]

2.3.8 Příklady využití embedded systémů v reálném světě

- Bankovní automaty,
- telefonní ústředny,
- mobilní telefony,
- vybavení pro počítačové sítě, jako jsou hardwarové routery a firewally,
- řídicí jednotky motorů automobilů a jejich přídatné systémy, jako je ASR, ABS apod.,
- domácí automatizace, jako jsou termostaty, otevírání vrat garáží, zabezpečovací technika,
- domácí vybavení, jako jsou mikrovlnné trouby, horkovzdušné inteligentní trouby, myčka nádobí, vysavače, TV, videorekordéry, DVD rekordéry apod.,
- herní videokonzole, jako jsou Playstation, Nintendo, GameBox apod.,
- inteligentní kalkulátory, přenosné počítače, PDA apod.,
- automatické navigační systémy letadel a raket,
- medicínské vybavení,
- programovatelné automaty PLC.

2.3.9 Současní světoví výrobci embedded systémů, jejich příslušenství a softwarového vybavení.

Současný trh s embedded systémy a jejich příslušenstvím je obsazen zejména firmami Advantech, Siemens, Compaq, Hewlett-Packard, IBM, AMD, Intel, VIA, Microsoft, Hitachi, Motorola, Nokia, Sony-Ericsson, Fujitsu.

Mezi majoritní firmy dodávající software pro embedded systémy patří Microsoft a Symbian. Samostatným členem software pro embedded systémy je volně šiřitelný operační systém Linux.

3 FUZZY LOGIKA A FUZZY MNOŽINY

3.1 Principy fuzzy logiky a fuzzy regulace

Základem fuzzy logiky byla práce polského matematika a logika J. Łukasiewicze, který se ve 30. letech minulého století zabýval teorií neurčitosti a vícehodnotovou logikou. Na jeho práci pak významně navázal v 60. letech minulého století profesor L. A. Zadeh, který tuto vědní disciplínu rozvinul ve vědní obor a umožnil tak vzniknout fuzzy logice.

Fuzzy logika je matematická disciplína, která si získala značnou popularitu koncem 80. a na začátku 90. let minulého století. Hlavním zdrojem jejího úspěchu je to, že fuzzy logika umožňuje zahrnout do svého „zorného pole“ nepřesnost a poměrně jednoduchým způsobem pracovat s významy slov přirozeného jazyka, který patří mezi nejdůležitější součásti lidského života. Proto fuzzy logika zasahuje do mnoha oblastí lidské činnosti a stále se nacházejí další možnosti jejího využití.

Jak může tedy fuzzy logika tak úspěšně a přesně fungovat? Je to tím, že využívá nepřesně charakterizované expertní znalosti a tyto promítá do svého nejistého a nepřesného chápání problému, a tím se dostává k výsledku rychleji než exaktní vědní obory.

3.2 Použití fuzzy logiky při regulaci

Myšlenka použít fuzzy logiku k řízení průmyslových procesů pochází od L. A. Zadeha a byla dále rozpracována S. Assilianem a E. H. Madanim z Londýnské univerzity. Postup, který vyvinuli a navrhli, se nazývá fuzzy regulace. Tato je přímým opakem regulace klasické.

Klasickou regulaci lze uskutečnit pouze za předpokladu, jestliže známe matematický popis systému/procesu, který chceme regulovat. Teorie klasické regulace je velmi dobře rozvinutá disciplína, která nabízí spolehlivá řešení, a to zejména tehdy, je-li proces přesně popsán pomocí lineárních diferenciálních rovnic s konstantními koeficienty.

Ovšem, známe-li podmínku a máme k dispozici výsledky pozorování systému, pak můžeme lehce usoudit, co máme dělat, tj. jaký má být akční zásah. Tento proces usuzování představuje základ fuzzy regulace a různých dalších aplikací včetně empirického rozhodování. A právě fuzzy logika poskytuje prostředky jak takovou regulaci realizovat.

3.3 Fuzzy logika

Byla poprvé použita při automatizovaném řízení podzemní dráhy v Japonsku, pro řízení soustavy rychlovýtahů ve výškových mrakodrapech, různých pozemních strojů a v celé řadě dalších aplikací.

Ovšem fuzzy logiku aplikují i jiné firmy do svých oborů činnosti, např. americká firma Boeing využívá tuto logiku ve svých leteckých navigačních systémech a systémech řízení letu, japonská firma Sony využívá tuto logiku do spotřební elektroniky a Matsushita Electronic Works využívá tuto logiku na vytváření inteligentních řídicích systémů pro zabezpečení sítí, osobní hygienické systémy (jako jsou inteligentní holicí strojký, automatizované toalety apod.) a například ještě inteligentní řídicí systémy pro automobilový průmysl. [16]

3.4 Využití fuzzy logiky

Fuzzy regulace není jedinou oblastí, kde se fuzzy logika uplatňuje. Další významnou oblastí je rozpoznávání obrazu a zvuku. Teorie fuzzy logiky a fuzzy množin se uplatňují také v databázových systémech, v nichž umožňuje velmi výkonné vyhledávání, mimo jiné vyhledávání informací na základě nepřesně specifikovaných charakteristik.

3.5 Fuzzy regulátory

Jsou velmi robustní, tj. necitlivé vůči změnám pracovních podmínek. Dalším důležitým rysem fuzzy regulátorů je to, že jsou ve srovnání s klasickými regulátory obvykle levnější. Z výše zmíněných vlastností plyne, že fuzzy regulátor je takřka optimální.

3.6 Princip fuzzy regulace

Fuzzy regulace je ve své podstatě aplikací teorie přibližné dedukce na řízení technologických procesů. Fuzzy regulátor lze zapojit do uzavřené zpětnovazební smyčky, kdy pouze nahrazuje klasický regulátor. Všechny výše uváděné výhody jdou zde zachovány.

Ještě významnější možností je zapojit jej do složitějšího hierarchického řídicího systému, jehož prvky jsou realizovány různými algoritmy. Podle konkrétní situace jsou jeho prvky buď samostatné fuzzy regulátory resp. klasické regulátory, popř. adaptivní regulátory.

Hlavním řídicím systémem, který zapojuje jednotlivé řídicí prvky, je fuzzy regulátor, který je pak spíše v roli expertního systému.

Fuzzy regulátor pracuje na odlišném principu. Jeho hlavní myšlenka spočívá v předpokladu, že matematický popis systému nemusí být v praxi vždy znám, avšak je znám způsob, jak řídit – regulační strategie. Tato strategie je popsána pomocí množiny výroků JESTLIŽE-PAK v té podobě, jak je používá člověk.

Protože jediným požadavkem návrhu fuzzy regulátoru je znalost regulační strategie, která navíc může být poměrně hrubá, nemusíme hledat matematický popis regulovaného procesu. To je jedna z hlavních výhod fuzzy regulátorů, protože odpadá obtížná a drahá identifikace procesu, která navíc nemusí vždy vést k uspokojivým výsledkům.

3.6.1 Základní postup při aplikaci fuzzy regulace

Stanovíme, jaké budou závisle a nezávisle proměnné (akční zásah). To závisí na tom, zda regulace probíhá v uzavřené zpětnovazební smyčce, o jaký typ fuzzy regulátoru se jedná, popř. na dalších speciálních podmínkách, v nichž fuzzy regulátor aplikujeme.

Rozhodneme o typu fuzzy regulátoru a způsobu přibližné dedukce, tj. zda použijeme fuzzy logickou dedukci nebo fuzzy aproximaci funkce. Sestavíme bázi znalostí, tj. na základě expertní informace popíšeme strategii řízení pomocí jednoho nebo více jazykových popisů (množin fuzzy pravidel typu JESTLIŽE-PAK). U všech proměnných pak určíme jejich jazykový kontext.

3.6.2 Typy základních regulátorů a jejich fuzzy variant

Základní metodou řízení v teorii regulace je řízení podle odchylek. Konkrétněji se dá postupovat takto, když bude w_t požadovaná hodnota, x_t je výstup řízeného procesu a u_t je akční zásah v okamžiku t , pak odchylka je $e_t = w_t - x_t$ a změna odchylky je $\Delta e_t = e_t - e_{t-1}$, druhá diference odchylky je $\Delta^2 e_t = \Delta e_t - \Delta e_{t-1}$.

Analogicky jako u klasické regulace lze v případě regulace v uzavřené zpětnovazební smyčce rozlišit fuzzy regulátory typu P, PI, PD a PID. Z matematického hlediska je fuzzy regulátor nelineární funkce, která je určena pomocí fuzzy pravidel typu JESTLIŽE-PAK. Závisle na charakteru a počtu nezávisle i závisle proměnných se dají analogicky rozlišit 4 základní typy fuzzy regulátorů.

a) P-regulátor

- klasický:

Jedná se o nejjednodušší regulátor, a je proto určen funkcí

$$u_t = C_P(e_t) = K_P \cdot e_t \quad (1)$$

kde K_P je konstanta (proporcionální zesílení).

- fuzzy:

U fuzzy P-regulátoru je akční zásah určen přímo odchylkou:

$$\text{JESTLIŽE } e \text{ je } A_e \text{ PAK } u \text{ je } B_u \quad (2)$$

b) PD-regulátor

- klasický:

Je určen funkcí odchylky a její změny

$$u_t = C_{PD}(e_t, \Delta e_t) = K_P \cdot e_t + K_P \cdot e_t + K_D \cdot \Delta e_t \quad (3)$$

kde K_D je derivační konstanta a K_P je zesílení.

- fuzzy:

Akční zásah je určen přímo odchylkou a její změnou:

$$\text{JESTLIŽE } e \text{ je } A_e \text{ A } \Delta e \text{ je } A_{\Delta e} \text{ PAK } u \text{ je } B_u \quad (4)$$

c) PI-regulátor

- klasický:

Pokud se nenastavuje přímo akční zásah, ale jen jeho změna, pak platí

$\Delta u_t = u_t - u_{t-1}$ a funkce PI regulátoru je :

$$\Delta u_t = C_{PI}(e_t, \Delta e_t) = K_I \cdot e_t + K_P \cdot \Delta e_t \quad (5)$$

kde K_I je integrační konstanta a K_P je zesílení.

- fuzzy:

Zadává se pouze změna akčního zásahu. Tento typ se používá nejčastěji, protože pro člověka je snazší myslet jen na dvě nezávisle proměnné a určovat změnu akčního zásahu:

$$\text{JESTLIŽE } e \text{ je } A_e \text{ } \Delta e \text{ je } A_{\Delta e} \text{ PAK } \Delta u \text{ je } B_{\Delta u} \quad (6)$$

d) PID-regulátor

- klasický:

Jedná se o funkci odchylky, její změny a její druhé diference, jeho funkce je

$$\Delta u_t = C_{PID}(e_t, \Delta e_t, \Delta^2 e_t) = K_I \cdot e_t + K_P \cdot \Delta e_t + K_D \cdot \Delta^2 e_t \quad (7)$$

kde K_I je integrační konstanta, K_D je derivační konstanta a K_P je zesílení.

- fuzzy:

Je nejsložitější, ovšem nutný pro vysoce nelineární a nestabilní procesy. Hlavním problémem u tohoto fuzzy regulátoru je určení pravidel, neboť jejich počet může být dosti velký. V praxi lze často nahradit fuzzy PID regulátor kombinací fuzzy PI a PD regulátorů). Jeho fuzzy JESTLIŽE – PAK funkce vypadá takto:

$$\text{JESTLIŽE } e \text{ je } A_e \text{ } \Delta e \text{ je } A_{\Delta e} \text{ } \Delta^2 e \text{ je } A_{\Delta^2 e} \text{ PAK } \Delta u \text{ je } B_{\Delta u}. \quad (8)$$

3.6.3 Stanovení parametrů fuzzy regulátoru

Jedná se obecně o určení způsobu indiference a defuzzifikace, vytvoření báze znalostí a stanovení jazykového kontextu. Stanovení proměnných je však základní úlohou, ze které vychází celý další postup. Nezávislé proměnné budou odchylka e a její první popř. druhá diference. Závislé proměnné budou u (akční zásah) nebo jeho diference.

U odchylky je nutno specifikovat :

- nejmenší možnou a nejvyšší možnou mez $[e_{\min}, e_{\max}]$,
- změnu odchylky $[\Delta e_{\min}, \Delta e_{\max}]$,
- druhou diferenci $[\Delta^2 e_{\min}, \Delta^2 e_{\max}]$.

U akčního zásahu je nutno specifikovat:

- maximum a minimum $[u_{\min}, u_{\max}]$,
- změnu akčního zásahu $[\Delta u_{\min}, \Delta u_{\max}]$.

3.7 Fuzzy model řízeného procesu

Jde o jistý hybridní přístup, kdy se zcela neopouští požadavek znát nejprve model procesu a na základě něj odvozovat způsob jeho řízení. Ovšem jak model tak i řízení je konstruováno na základě výše popsaného fuzzy přístupu. Tím se zvyšuje pravděpodobnost, že se výsledný model bude chovat realističtěji, protože bere v úvahu složitost systému, z níž vyplývá nutnost pracovat s nepřesnou charakterizací situace. [15]

II. PRAKTICKÁ ČÁST

4 NÁVRH MONITOROVACÍ A ŘÍDICÍ SOUSTAVY

4.1 Vznik monitorovací a řídicí soustavy

Důvodem vzniku této monitorovací a řídicí soustavy je fakt, že na současném trhu s informačními a řídicími technologiemi existuje nedostatek řešení, která by mohla zastat výše uvedenou monitorovací a řídicí činnost pomocí mobilní sítě. Pokud takováto řešení existují, pak jsou nákladné a týkají se velkých průmyslových koncernů, které si mohou dovolit směřovat své aktivity tímto směrem.

Ale protože trh se nesešává jen s velkých průmyslových koncernů, ale také z firem menších a samozřejmě jednotlivých koncových zákazníků - je možné směřovat cíl této práce právě na uspokojení potřeb těchto lidí. Tito lidé ve většině případů nepotřebují regulovat složité průmyslové procesy, ale naopak jejich potřeby jsou orientovány spíše na regulaci malých tepelných soustav, jako je například byt, rodinný dům nebo chata.

Proto vznikla tato práce, která by měla dát těmto lidem šanci pořídit si relativně laciný regulátor, který dokáže regulovat výše uvedené tepelné soustavy a který by mohl být ovládán pomocí mobilního telefonu nebo sítě internet, což je v podstatě neustále a téměř kdekoli kde je dostupný signál mobilní sítě.

4.1.1 Podmínky pro realizaci

Protože v naší zemi existují již tři mobilní operátoři a pokrytí území našeho státu signálem je již na velmi dobré úrovni a internet je již součástí téměř každé domácnosti, je možné zahájit vývoj takového regulátoru, který by za pomoci již zavedené infrastruktury nabízené našimi mobilními operátory mohl zastávat výše uvedenou činnost – tedy regulovat tepelný systém a být přitom řízen na dálku.

Cenová dostupnost jednotlivých součástí monitorovací a řídicí soustavy je v současných podmínkách, které jsou na trhu s informačními a řídicími technologiemi, na relativně nízké úrovni, proto i tento faktor dává obsahu této práce smysl.

4.1.2 Shrnutí faktů a budoucí použitelnost modelu jako vzoru

Samotné úsilí, které je nutné vynaložit na vytvoření takové soustavy, je dle mého názoru v rozumné míře, protože i středně pokročilý vývojář či programátor, který má dostatečné

informace o soudobých technologiích, dokáže navrhnout, ověřit a realizovat takový monitorovací a řídicí systém během řádově několika měsíců. Tento můj názor potvrdilo zejména to, že jsem celou tuto práci vypracovával samostatně a jsem schopen určit časovou a finanční náročnost celého projektu.

Po zjištění tohoto faktu jsem se rozhodl přenést všechny zjištěné skutečnosti do své práce tak, aby tento projekt mohl být kdykoliv zopakován. Přesto zde uvádím pouze klíčové informace, které jsou zásadní pro celý projekt a vynechávám drobnosti, které by pouze zahlcovaly papír a samotnou opětovnou realizaci by nijak neurychlily.

Přestože tento projekt během svého vývoje prošel řadou změn a jistě ještě není dokonalý a bude obsahovat nějaké nejasnosti či zbytečně složité konstrukce, domnívám se, že se může stát startovní čarou pro finální řešení, které by mohlo být s úspěchem nasazeno na náš trh.

4.2 Popis technologického procesu

Jako technologický proces byla vybrána jednoduchá tepelná soustava. Cílem práce totiž není najít řešení regulace specifického a složitého procesu, ale naopak najít nejvhodnější a pokud možno co nejobecnější technologický proces, na jehož ukázce bude možno realizovat projekt s podstatně složitějším procesem.

4.2.1 Základní informace vybrané o tepelné soustavě

Protože se jedná o tepelnou soustavu, je důležité si uvědomit několik základních faktů, které mohou více o této soustavě prozradit. Jako první bych uvedl fakt, že regulace tepelných soustav je v současnosti dobře propracovaná a zvládnutá disciplína. Dále je známo mnoho postupů, jak si zjistit co nejvíce údajů o takovýchto soustavách a samozřejmě k této soustavě existuje velké množství reálných příkladů a možností využití, které se mohou stát budoucím pilířem kvalitního projektu, který již může mít i určitý obchodní úspěch.

4.2.2 Popis prostředí tepelné soustavy

V této diplomové práci se pracuje s tepelnou soustavou, která je uzavřená a měla by simulovat místnost. Z toho tedy vyplývá, že látka, která bude základem této soustavy, je

atmosférický vzduch. Vzduch je v podstatě směs plynů kyslíku, dusíku, oxid uhličitýho a dalších. Ve vzduchu se také nachází vodní pára.

Poměr základních plynů ve vzduchu je průměrně 20% O₂, 78% N₂ a 1% CO₂ a zbylé procento tvoří ostatní plyny – viz tabulka č. 5. Jelikož je vzduch relativně řádně prozkoumaný plyn, je možno na základě již známých skutečností s ním pracovat a podle jeho známého chování určit, jak se bude v dané soustavě chovat.

Tab. č. 4 – Složení atmosférického vzduchu

Složka	Objem (%)
N₂	78 %
O₂	21 %
Ar	0,9 %
Ne, He, Kr, Xe, H, Ra, NO_x, O₃, NH₃, CH₄ + průmyslové exhaláty, H₂O ve třech skupenstvích	0,1 %

Pozn.: Do 20 km nadmořské výšky se nachází 90 % atmosférických plynů, do 80 km nadmořské výšky se nachází zbylých 9 % a s narůstající nadmořskou výškou atmosféra řídne až plynule přechází v kosmický prostor (téměř vakuum).

Protože se jedná o uzavřenou soustavu, potřebuje i tato nějaký obal, který by ji oddělil od okolního prostředí. V této diplomové práci byl navržen jako obal materiál, který by byl co nejjednodušeji zpracovatelný a cenově přístupný – a to dřevo, přesněji dřevotříska.

Neboť dřevotříska je materiál uměle vytvořený, existuje dostatek materiálů, jak popsat její fyzikální vlastnosti a zejména chování v tepelných systémech. Jedná se v podstatě o slisované a nějakou ztužovací látkou zpevněné piliny, které mají tvar desky. Obvykle ještě bývá pokryta vrstvou dýhy, což je v podstatě směs umělých hmot a barviv. Tato vrstva

dává dřevotřískice vzhled a mění její tepelně izolační vlastnosti. Základní vlastnosti dřevotřísky se nacházejí v příloze PV.

Chování tepelné soustavy ovlivňují tepelné ztráty. Tyto tepelné ztráty představují zejména nezávislé a těžce předvídatelné události, které mohou nastat během existence takového systému. Mohou jimi být například netěsnosti obalového materiálu, netěsnosti u ventilač-

ních a prosvětlovacích otvorů a samozřejmě ztráty způsobené vedením tepla izolačním materiálem.

4.2.3 Použité topné těleso

Topné těleso využití v této soustavě bylo navrženo tak, aby bylo možno v rozumných časových intervalech sledovat chování tepelného systému. Reálný poměr výkon (kW) : objem prostředí (m³) nebylo vhodné dodržet. Topným tělesem se pro tuto práci stala 60 W žárovka, jejíž použití umožnilo velmi rychlé změny stavu simulované regulované soustavy.

4.3 Návrh monitorovací a řídicí soustavy

Monitorovací a řídicí soustava byla navržena tak, aby umožnila nejvyšší možnou kvalitu řízení a monitorování navrhované simulované tepelné soustavy. Je složena z Biscuit PC počítače, pomocné elektroniky, teplotních čidel a software, který s těmito součástmi spolupracuje. Jednotlivé části spolupracují tak, aby docházelo k minimálním časovým zpožděním

a co nejkratším reakčním dobám této soustavy.

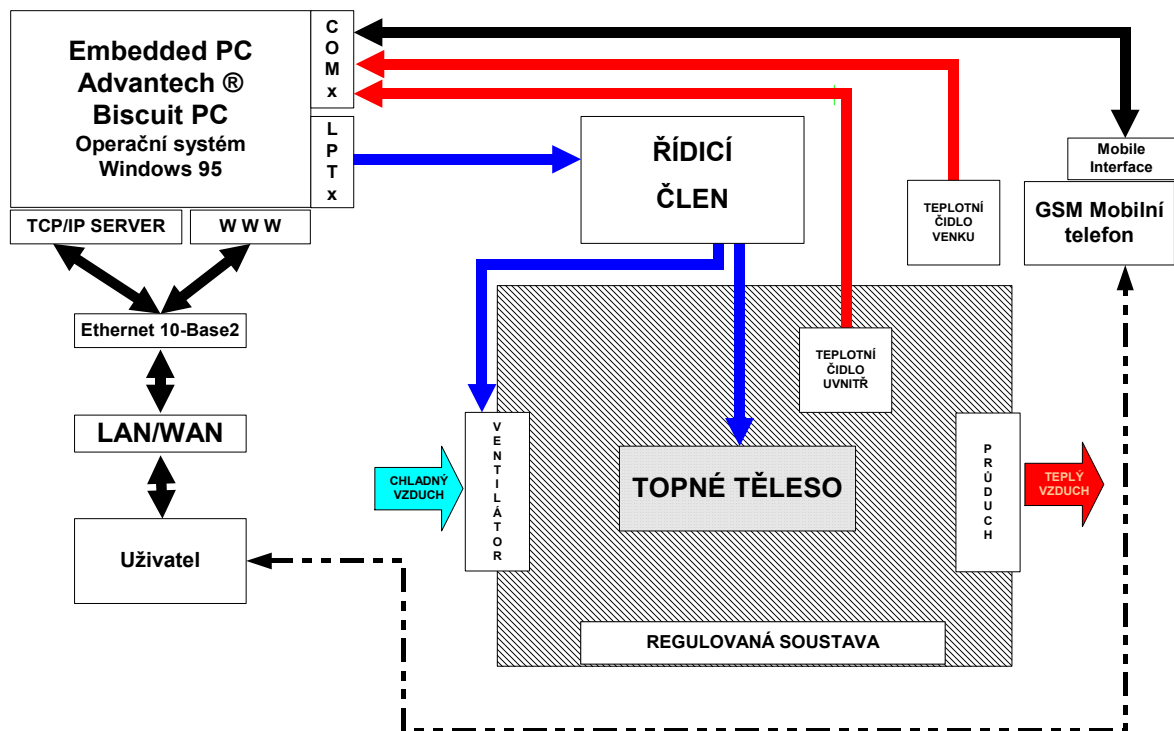
Jako základ celého řídicího a monitorovacího systému byl zvolen Biscuit PC, který zastane všechnu práci týkající se monitoringu a regulace. Dále byl použit GSM modul pro využití GSM sítě, do soustavy bylo nutno implementovat teplotní čidla vhodná pro výše navrženou tepelnou soustavu a navrhnout a sestavit pomocná elektronická zařízení umožňující řízení připojených elektrických zařízení. Nakonec bylo potřeba toto všechno zastřešit návrhem software, který by výše zmíněnou soustavu dokázal ovládat a řídit.

K tomu, aby všechny činnosti spojené s realizací soustavy a systému byly koordinované a efektivní, bylo nutné vytvořit základní principiální schéma, které by teoreticky obsahlo základní prvky tohoto systému a umožnilo tím provést jeho realizaci.

4.3.1 Schéma soustavy a monitorovacího a řídicího systému

Ve schématu se počítá s využitím výkonného Biscuit PC, který bude schopen zvládat nejen provoz regulátoru, ale také WWW a GSM serveru. Toto schéma se nachází na obrázku č.17. Je nutno podotknout, že níže uvedené schéma bylo vytvořeno v produktu Microsoft® Visio 2000. [30]

4.4 Schéma řešení



Obr. č. 17 – Modulární schéma řešení

4.4.1 Popis schématu

Klíčovými prvky tohoto schématu jsou Embedded PC Advantech® Biscuit PC a Regulovaná soustava.

Biscuit PC zde hraje úlohu multifunkčního regulátoru. Jelikož se jedná o výkonné zařízení, tak v něm bude také zabudován GSM server, WWW server a TCP/IP telnet server. Tyto servery budou nabízet rozhraní, pomocí kterého bude možno řídit regulovanou soustavu.

K Biscuit PC bude pomocí sériového RS-232 rozhraní připojen mobilní telefon a vnější a vnitřní teplotní čidlo. Pomocí paralelního rozhraní IEEE 1284 bude připojen řídicí člen, který umožňuje řízení ventilátoru a topného tělesa. Biscuit PC bude připojen k Ethernet síti pomocí Ethernet 10-Base2 rozhraní.

Regulovaná soustava bude mít k dispozici topné těleso, které bude prostor regulované soustavy vytápět. Ke kontrole stavu soustavy v ní bude zabudováno teplotní čidlo. Pro simulaci poruchy a pro rychlejší vychlazení soustavy bude k dispozici ventilátor.

Uživatel bude regulovaný systém ovládat buďto mobilním telefonem nebo WWW prohlížečem či telnetovým klientem. Komunikace mezi uživatelem a regulátorem bude probíhat přes GSM nebo ethernet.

Regulátor bude ovládat regulovanou soustavu pomocí proprietárního hardware – ve schématu je tento označen jako „Řídicí člen“. Řídicí člen má za úkol pomocí řídicího signálu přicházejícího z Biscuit PC spínat a rozepínat relé topného tělesa, ventilátoru. Dále má za úkol přepínat bránu děliče COM portů.

5 POUŽITÝ HARDWARE

V předešlé části jsem zmínil vybrané schéma, dle kterého je vytvořen reálný model regulované soustavy. V této části bych se chtěl věnovat důkladnějšímu popisu jednotlivých pevných součástí schématu.

5.1 Embedded PC Advantech® Biscuit PC

Toto zařízení je klíčové pro celé schéma, protože v něm bude pracovat program, který bude realizovat regulátor, GSM, Telnet a WWW servery. Protože požadavky na toto zařízení jsou pevně dané – bude to Embedded PC Advantech® Biscuit PC mající sériové, paralelní a ethernet rozhraní.

Přesný model použitého Biscuit PC je deskový počítač – PCM-5822. Toto zařízení se objevilo ve firmě Datasys, s.r.o. jako podpora pro specifické projekty, ve kterých jsou využívány embedded technologie. [34]

Konkrétní Model PCM-5822 co je použit v této diplomové práci je vybaven 128 MB SD-RAM a 128 MB CF-CARD flash diskem od firmy San-Disk.

Specifikace této CF-CARD datové karty je:

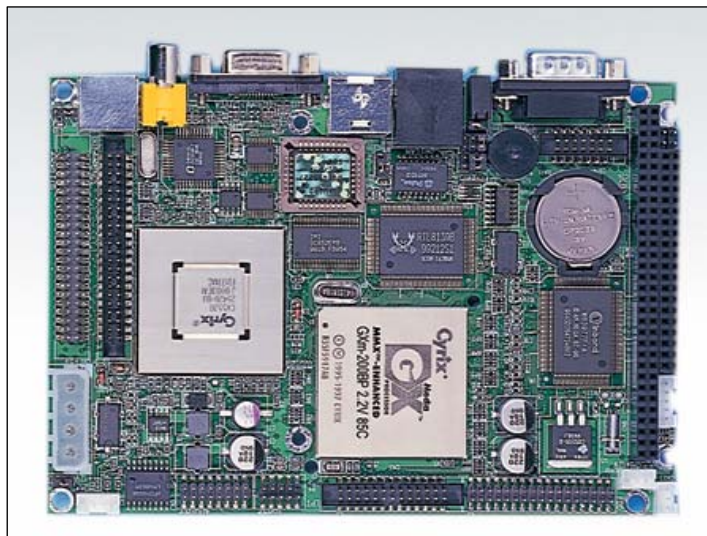
- Výdrž / zátěž : 1 000 000 programovacích/mazacích cyklů,
- Provozní napětí: 3.3V/0.5V,
- Provozní teplota: 0 °C – 700 °C (32 °F – 158°F),
- Doporučené použití: Záloha dat, sdílení dat, úložný prostor pro fotografie a datové soubory,
- Velikost (š x d x v): 42,8 x 36,4 x 3,3 (mm),
- Váha: 11,4 g,
- Obecné vlastnosti:

Ultra rychlá přístupová doba,

Nízký poměr výkon vs. cena,

Nízký odběr elektrické energie,

Vyhovuje stanoveným průmyslovým standardům.



Obr. č. 18 – Vzhled desky Biscuit PC Advantech® PCM-5822

5.1.1 Základní popis PCM-5822-D0A1

- CPU na desce AMD Geode™ processor, GXLV-200 MHz processor ,
- Vyrovnávací paměť (Cache Memory) dvou úrovněv,
- Čipová sada: AMD CX 5530,
- BIOS AWARD® 256 kbit Flash BIOS,
- Systémová paměť 1 x SDRAM SODIMM, Max. 256 MB,
- Řízení spotřeby (Power Management),
- SSD Supports CompactFlash® card,
- WatchDog Timer softwarový povolen/zakázán 1.6 sec., průměrně 65 sec.,
- Rozšiřující rozhraní PC/104,
- Baterie Lithium 3 V/196 mAh.

5.1.2 Parametry vstupně výstupních zařízení

- I/O Interface 1 x EIDE (Ultra DMA33), 1 x FDD, 1 x K/B, 1 x Mouse, 1 x RS-232/422/485, 1 x RS-232, 1 x LPT,

- USB 2 x USB, USB 1.0 kompatibilní,
- Zvuk AMD CX 5530, AC'97 2.0 kompatibilní,
- IrDA 115 kbps, IrDA 1.0 kompatibilní,
- Rozhraní Ethernet:
Chipset RealTek® 8139C, rychlost 10/100 Mbps (dle potřeby), rozhraní 1 x RJ45, standard IEEE 802.3u 100Base-T Fast Ethernet kompatibilní, vestavěná Boot ROM ve flash BIOS.

5.1.3 Zobrazovací zařízení

- Čipová sada AMD CX 5530,
- Grafická paměť 1 - 4 MB UMA (sdílená paměť),
- Rozlišení až 1024 x 768 na 24 bpp - neprokládaný režim CRT ,
nebo 1024 x 768 na 18 bpp – prokládaný režim TFT LCD display (3.3V LCD),
- LCD rozhraní 18-bit TTL pro TFT LCD ,
- TV-out Chromtel® CH 7003B podporující NTSC i PAL,
- TV formát – poskytuje kompozitní S-Video a SCART.

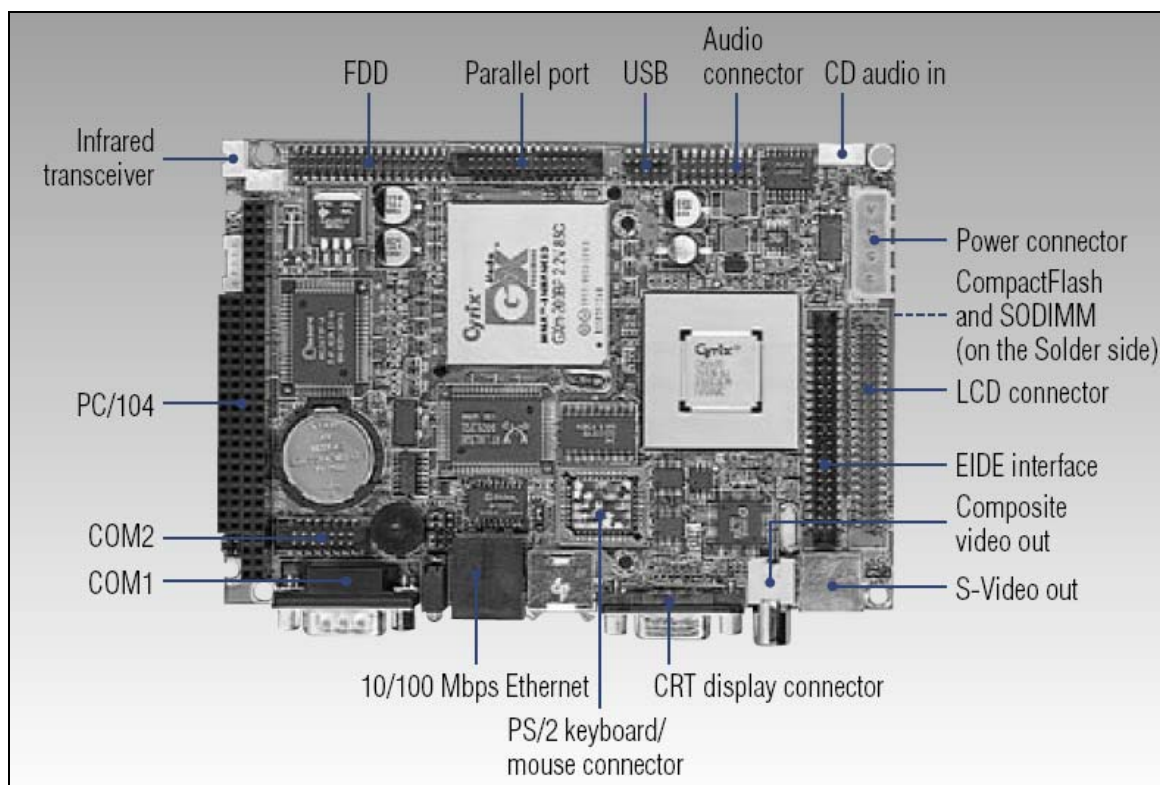
5.1.4 Mechanické parametry a pracovní podmínky

- Rozměry (d x š) 146 x 102 mm (5.7" x 4"),
- Váha 0.75 kg (celé balení),
- Pracovní teplota 0 ~ 60° C (32 ~ 140° F),
- Pracovní vlhkost prostředí 0% ~ 90% (relativní vlhkost, nekondenzující).

5.1.5 Napájení

- Napájecí napětí +5 V \pm 5% (stejnoseměrných),
- Odběr maximálně 4 A na +5 V.

5.1.6 Rozložení součástí na základní desce



Obr. č. 19 – Rozložení součástí na základní desce Advantech ® PCM-5822

5.2 Uživatelský počítač

Jako desktop PC bude použit osobní počítač, který mám přidělen firmou Datasys, s. r. o. Jelikož se jedná o vývojový počítač, musel jsem opět požádat o povolení tento počítač využít pro potřeby mé diplomové práce. Jelikož má tento počítač velmi dobré parametry, bude velmi vhodnou součástí pro tuto měřicí a řídicí soustavu. Navíc má nainstalován všechen potřebný software, který je potřebný pro návrh, vývoj a ladění výše uvedeného měřicího a regulačního zařízení.

5.2.1 Základní hardwarové parametry

- Základní deska obsahující jeden Socket 478 a tři paměťové sloty pro DDR paměti (do každého z nich může být vložena DDR paměť o maximální kapacitě 512 MB),
- 5 x PCI slot, 1 x AGP(8x) slot, jeden sériový port, jeden paralelní port, 4 x port USB 2.0 a ethernetový adaptér pro 10BaseT (100 Mbps),
- CPU Intel Pentium IV, taktovací frekvence 2,4 GHz,

- RAM paměť v provedení DDR o velikosti 512 MB,
- HDD s podporou Ultra ATA, 7200 ot./min, kapacita 120 GB,
- DVD-ROM s podporou Ultra ATA (50x čtení CD, 2x čtení DVD),
- Grafická karta v provedení AGP, NVidia Ge-Force 4-MX,
- PS/2 klávesnice a PS/2 optická myš,
- Monitor LCD, úhlopříčka 19“.

5.2.2 Operační systém a instalovaný kancelářský a vývojový software

- Microsoft® Windows 2000™ Advanced Server + IIS,
- Microsoft® Visual Studio™ 6.0,
- Microsoft® Embedded Visual Tools™ 3.0,
- Microsoft® Platform Builder™ 2.11 a 3.0,
- Microsoft® Office™ 2000,
- Microsoft® Visio 2000,
- Apache WWW server + PHP interpreter.

Myslím si, že tento počítač velmi dobře vyhovuje požadavku na vývojový počítač, protože jeho výkon a instalovaný software umožňuje jeho využití ve všech fázích projektu. Od návrhu, realizace až po testování.

5.3 Řídicí člen

Řídicí člen byl navržen s ohledem na realizační schéma a další dostupný hardware. Neboť jsem již při vytvoření schématu celé řídicí a regulační soustavy předpokládal existenci dodatečného řídicího členu, mohl jsem se nyní ve fázi návrhu potřebného hardware začít věnovat také návrhu této velmi důležité součásti. Jeho úlohou je zprostředkovat akční zásah regulátoru, a to tak, že bude spínat nebo vypínat tepelný zdroj.

Dále je zde možnost, že pro potřeby simulace funkčnosti regulátoru bude třeba vytvářet poruchy, které by mohly simulovat reálné výkyvy teploty, nebo rychle odvětrat prostor regulované tepelné soustavy. Toto bude zprostředkováno pomocí ventilátoru.

5.3.1 Relay desk v1.0

Toto zařízení bude ovládáno z PC pomocí signálů paralelního portu a bude spínat potřebné elektrické součásti regulované soustavy, tedy topné těleso a ventilátor.

Během návrhu jsem musel vyřešit problém, jak výkonné má být relé, které bude spínat výše uvedené elektrické součásti. Protože jsem potřeboval, aby relé bylo dostatečně výkonné pro spínání silného zdroje tepla (pro rychlejší průběh regulace), vybral jsem relé **Schrack 210012**, které dokáže spínat až 250 V střídavého napětí a umožňuje průchozí proud do 10 A. Maximální přípustný výkon spínaného obvodu je tedy 2,5 kW.

Odpor vinutí cívky tohoto relé je přibližně 630 Ω . Spínací napětí je 12V stejnosměrného napětí. Pro vypnutí musí být napětí menší než 1,7 V. Od 1,7 V do 3 V je nedefinované pásmo, kdy stav relé není přesně definovaný. Pro ovládání relé bude tedy použito 0V pro vypnutí a 12V pro sepnuto.

Při ovládání relé je nutno vzít na potaz fakt, že maximální proud, který dokáže dodat paralelní port, je 20 mA. Pro spínání relé by to mělo stačit, ovšem pro jistotu jsou použity tranzistory, aby relé od spínacího obvodu oddělily. Provedení reléové desky navrhuji tak, aby na jednom plošném spoji byly umístěny dvě relé, z nichž každé bude řízeno tranzistorem. Tranzistory již budou ovládány pomocí výstupů paralelního portu. Při nastaveném výstupu paralelního portu relé spíná, při nenastaveném výstupu relé vypíná.

5.3.2 Com Switch v1.0

Důvod vzniku Com Switch v1.0 zařízení je ten, že Biscuit PC Advantech ® PCM-5822 podporuje samo o sobě pouze 2 sériové porty a ze zadání vyplývá požadavek na použití 3 sériových zařízení – vnitřní čidlo, vnější čidlo a GSM zařízení.

První zařízení – GSM mobilní telefon - tedy bude zapojeno na COM1. Zbývá dvě sériová teplotní čidla budou spojena s COM2 pomocí Com Switch. Zařízení Com Switch tedy bude umožňovat práci s oběma čidly.

Navržená architektura Com Switch v1.0 neumožňuje paralelní práci všech připojených zařízení, ovšem to není nezbytně nutné, neboť vyčtení informace z teplotního čidla trvá řádově vteřiny a přepnutí okruhu trvá řádově několik milisekund. A protože se jedná o

regulaci tepelné soustavy, je jasné, že zpoždění v řádu vteřin nemůže hrát zásadní roli v kvalitě regulačního procesu.

Funkcionalita zařízení je v podstatě jednoduchá. Rozborem činností jednotlivých sériových zařízení jsem došel k závěru, že pro správnou funkcionalitu připojených zařízení stačí propojit pouze tyto signály sériového portu TX, RX, RTS, CTS a samozřejmě GND. Toto propojení se provede přes zem desky Com Switch v1.0. Logika přepínání jednotlivých zařízení je realizována pomocí integrovaných obvodů.

Vstupem Com Switch v1.0 jsou jednotlivá připojená zařízení a kombinace bitů pro nastavení brány - signál z paralelního portu. Výstupem je jeden sériový port, který se propojí s konektorem na Biscuit PC. Při nastaveném výstupu se u daného konektoru rozsvítí signální LED zelené barvy.

Integrované obvody, které jsou použity v tomto zařízení:

- 74138 – 3 na 8 adresový dekodér/demultiplexer,
- 74153 – 8 na 2 adresový kodér/multiplexer,
- 7406 – 6 x INV,
- 7408 – 4 x 2 AND.

Tab. č. 5 – Adresování zařízení pomocí Com Switch v1.0

Cílové zařízení	Adresa (binární/dekadická)
1	00 / 0
2	01 / 1
3	10 / 2
4 (nedefinované)	11 / 3

5.3.3 Teplotní čidlo TM fy PaPouch™ Elektronika

Protože problematika snímání teploty je již natolik složitá, že vytvoření teplotního čidla by zabralo příliš mnoho času, rozhodl jsem se najít u současných dodavatelů měřicí techniky takové čidlo, které by odpovídalo mnou navrženým parametrům. Takové čidlo jsem našel – jedná se o teplotní čidlo TM od firmy PaPouch™ Elektronika, které se dá připojit na sériový port a výsledkem jeho měření je přímo ASCII řetězec zapsaný na sériový port, ze kterého se dá změřená teplota lehce získat. [31]

Formát řetězce, který vrací teplotní čidlo, je tento „+/- xxx.x C“, kde xxx.x je číslo s jedním desetinným místem a dle potřeby doplněno na délku tří celých číslic nulami. Takže ukázková návratová hodnota pro +19,6 °C bude „+019.6 C“. Teplota je uvedena ve stupních Celsia.

Čidlo se aktivuje tak, že se nastaví signál DTR a do 1 sekundy čidlo změří první teplotu a pak již periodicky, dokud je nastavený signál DTR, vrací po 10 sekundách aktuální údaje o teplotě pomocí signálu RX. Čidlo vrací aktuální teplotu z místa, kde se nachází pouzdro s měřicím obvodem (konec kabelu). Umělým nastavováním a nulováním signálu DTR je možno dosáhnout rychlejšího měření, avšak experimentálně jsem dokázal, že nelze měřit rychleji než 5x za sekundu.

Teplotní rozsah tohoto čidla je v rozmezí -55 °C až +125°C. Přesnost měření je +/- 0,5 °C. Konfigurace RS232 portu pro použití s tímto čidlem:

- Rychlost rozhraní je 9600 bps,
- datových bitů 8,
- žádná parita,
- 1 stop bit. [31]



Obr. č. 20 – Teplotní čidlo TM od firmy Pa-Pouch® Elektronika

5.3.4 Mobilní telefon Siemens M75

Mobilní telefon, který jsem vybral pro realizaci své diplomové práce, je Siemens M75, tedy telefon série 75. Tento telefon odpovídá všem základním požadavkům na mobilní telefon, které byly uvedeny ve schématu.

Siemens M75 má k dispozici grafický display o rozlišení 132x172 pixelů. Je vybaven možností datových přenosů GPRS 10/4. Dále má ve výbavě Bluetooth™, IRDA, slot pro RS-MMC a samozřejmě rozhraní pro připojení datového kabelu. Tento telefon je vybaven Li-ION baterií, která tomuto telefonu umožňuje pohotovostní dobu o délce až 10 dní, délku

nepřetržitého hovoru zhruba 4 hodiny nebo odeslání a příjmu několika stovek SMS zpráv. [35]



Obr. č. 21 – Siemens M75

Co se týče požadovaných vlastností, tak tento mobilní telefon krom mnoha různých vylepšení a užitečných vlastností, obsahuje také požadovaný hardwarový modem, podporující AT+ příkazy. Podporovaný formát pro příjem, čtení a odesílání SMS je PDU – kódování 7 z 8, 8 bitů, 16 bitů UNICODE.

5.3.5 Datový sériový kabel DCA-500

Pro použití v mé diplomové práci se tento kabel DCA-500 hodí nejvíce, neboť podporuje širokou škálu mobilních telefonů Siemens řady x55, x65 a x75. Jeho maximální přenosová rychlost je 115200 bps. Pro použití s mobilním telefonem M75 má tento datový kabel následující nastavení:

- Rychlost rozhraní je 115200 bps,
- datových bitů 8,
- žádná parita,
- 1 stop bit.



Obr. č. 22 – Kabel DCA 500

6 POUŽITÝ SOFTWARE

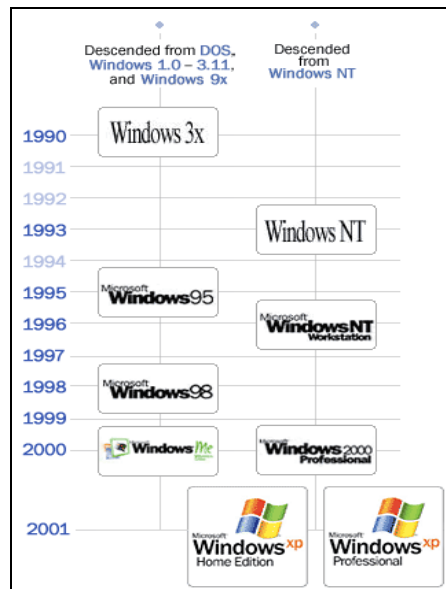
Pro přehlednější uspořádání celků této kapitoly budu dělit kapitoly nejprve na operační systém a vývojové prostředí, abych čtenáře uvedl do této problematiky a až následně se budu věnovat mnou psanému aplikačnímu software.

6.1 Operační systém v Biscuit PC

Jako operační systém byl vybrán Windows 95 od společnosti Microsoft®. Tento operační systém je v současné době zastaralý, ale vzhledem k nárokům na místo pro instalaci je velmi vhodný.

System Windows 95® byl poprvé uveden na trh v roce 1995. Byl to následník tří již existujících, v té době obecně používaných operačních systémů od společnosti Microsoft, a to Windows 3.1®, Windows for Workgroups® a MS-DOS®.

Windows 95® má v sobě integrovaný 32-bitový TCP/IP protokol (Transmission Control Protocol/Internet Protocol) pro podporu sítě Internet, podporu vytáčeného připojení sítě a využívá technologii Plug-and-Play. Dále má tento operační systém rozšíření multimediální schopnosti, zlepšený výkon a integrované síťové rozšíření. System je připraven pro 32 bitové počítače. [38]



Obr. č. 23 – časová linie vývoje operačních systémů od společnosti Microsoft

Instalace systému Windows 95 v Biscuit PC ® je minimální. Obsahuje pouze nejnútnejší součásti. Navíc jsou tam pouze přidány některé chybějící komponenty COM (Collaborate Object Model), vylepšený TCP/IP protokol a některé GDI (Graphics Desktop Instruments) prvky.

V této instalaci systému Windows 95 jsou zahrnuty také softwarové ovladače pro Advantech ® Biscuit PC PCM-5822, jako ovladače grafické karty, síťové karty a zvuku. Celková velikost instalace je zhruba 55 MB.

6.2 Apache WWW server

Na tomto serveru je nainstalován Apache WWW server, verze 1.4. Kvůli tomuto WWW serveru bylo nutno na operačním systému provést určité změny týkající se nastavení sítě, jako pevná IP adresa, upgrade TCP/IP protokolu a modernější COM objekty.

Tento WWW server má integrovanou podporu aktivních serverových skriptů PHP – a to verze 4.1. Instalace serveru zabírá kolem 14 MB.

6.3 Vývojové prostředí

Jako vývojové prostředí byl vybrán program společnosti Microsoft® Visual Studio 6.0, které je přímo připraveno k tomu, aby se v něm daly vyvíjet součásti operačního systému Microsoft® Windows 95.

Tyto součásti mohou být jak systémové knihovny a programy, ale samozřejmě také uživatelské programy. Vývojové prostředí je založeno na propracovaném GUI (Graphical user interface) a je koncipováno jako vizuální vývojový nástroj.

Podporovaný programovací jazyk je C/C++ s možnou podporou Microsoft® Foundation Classes – což jsou předvytvořené objekty, které zjednodušují práci s celým operačním systémem. Mají nevýhodu v tom, že integrace součástí MFC je realizována do několika podpůrných souborů o relativně velké velikosti, řádově jednotky MB, tedy program v psaný v objektovém MFC C++ je sice zdánlivě méně náročný na prostor a paměť, ale nutně vyžaduje koexistenci dynamických knihoven, které obsahují součásti, na které se daný program odkazuje.

Jeho hlavní součásti jsou:

- Návrhář projektů,

- editor jazyka C++ s grafickým interaktivním prohlížečem struktury programu/projektu,
- editor zdrojů programu, včetně grafického prostředí pro návrh dialogů a ovládacích prvků,
- editor textů.

Takové prostředí se dá nazvat výkonným, neboť obsahuje právě návrhář projektů, ve kterém se dá velmi jednoduše vytvořit nový projekt. Tento návrhář vytváří automaticky kompletní strukturu základních souborů potřebných pro daný projekt. Vývojář tedy nemusí psát velké množství textů, které jsou v podstatě pro každý projekt stejné a vývoj programu se značně urychlí. Projekt by měl být chápán jako soubor všech potřebných zdrojových souborů k vytvoření daného programu.

V současnosti je již k dispozici Microsoft Visual Studio .NET, moderní objektové vývojové prostředí, které umožňuje RAD (Rapid Application Development), což je technologie urychlující vznik kódu programu. Rovněž jeho interakce se součástmi operačního systému je na vyšší úrovni a má velké množství předpřipravených šablon programů. [36]

6.4 Vlastní software - Regulátor

Regulátor zabezpečí řídicí a monitorovací činnost, která se váže k výše uvedené regulované soustavě. Je napsán jako program v jazyce C++. Překlad bude upraven tak, aby linker přizpůsobil kód pro rychlejší běh a aby byly zarovnávány struktury.

Cílový typ užití znakové sady aplikace bude standardní ANSI. Nevýhodou je, že kód není možno přizpůsobit pro různé světové jazykové mutace, ovšem na druhou stranu je kód v ANSI menší a není nutno převádět znakové řetězce z BYTE (8 bitů) na WORD (16 bitů) – jak je to u varianty UNICODE. Navíc varianta ANSI nepotřebuje žádné speciální knihovny, neboť ANSI knihovny jsou standardně součástí Windows.

6.4.1 Typ regulátoru

Regulátor je navržen jako jednoduchý fuzzy-regulátor. Tento se rozhoduje na základě rozdílů teplot mezi aktuální a žádanou takovým způsobem, že na základě velikosti rozdílu (regulační odchylky) vybere správný topný režim. Regulační průběh je stanoven tak, aby měl pulsně-šířkovou charakteristiku, jeho topný výkon bude určen počtem topných úseků

z jedné minuty. Celkový počet časových úseků je stanoven na 10 – tedy každý jeden díl má 6 vteřin.

Výkon „nejmenší“ je 1 díl z minuty (topit 6s ze 60s), a tak dále, až „největší“ bude 10 dílů z minuty, tedy (60s ze 60s). Tedy z fuzzy pohledu to budou fuzzy výrazy od „netop“ ... přes “top málo“ ... až „top naplno“. Základní blokové schéma regulátoru je uvedeno v příloze P VIII.

6.4.2 Konfigurace regulátoru

Regulátor je z matematického hlediska navržen pomocí fuzzy podmínek **JESTLIŽE-PAK**, pomocí kterých se pak pokusí určit výslednou reakci na základě vstupních dat. Základní předpoklad je ten, že regulovaný systém je pomalý, proto se jeho teplota mění s narůstajícím časem při konstantním topném výkonu pomalu.

Jelikož potřebuji vytvořit takový fuzzy model, který by šel velmi rychle převést do programu, je nutné začít takto. Zvolím si maximální a minimální použitelnou hodnotu odchylky (e):

Minimální: $e_{\min} = 0,1$

(Při menších odchylkách bude topný výkon nulový)

Maximální: $e_{\max} = 3$

(Při větších odchylkách bude topný vždy výkon maximální)

Dále je nutno stanovit si oblast, ve které se bude vyskytovat Y , tedy řídicí zásah:

$Y_{\min} = 0$ (Tento řídicí zásah říká, že se nebude topit vůbec – 0%)

$Y_{\max} = 9$ (Tento řídicí zásah říká, že se bude topit 10 z 10 časových úseků – 100%)

Nyní jde pouze o to zvolit si vhodné jazykové operátory, které se přiřadí do fuzzy podmínek **JESTLIŽE – PAK**. Rozdíl žádané a aktuální teploty (Δt) si označím jako e , výsledný akční zásah, tedy topný výkon, si označím jako Y . Z tohoto pak vyplývá tato sada podmínek:

JESTLIŽE $e =$ „nulové“ **NEBO** $e <$ „nulové“ **PAK** $Y =$ „nulové“

JESTLIŽE $e \leq$ „velmi výrazně malé“ **A** $e >$ „nulové“ **PAK** Y je „malé“

JESTLIŽE $e \leq$ „velmi značně malé“ **A** $e >$ „velmi výrazně malé“ **PAK** Y je „zhruba malé“

JESTLIŽE $e \leq$ „značně malé“ **A** $e >$ „velmi značně malé“ **PAK** Y je „velmi zhruba malé“

JESTLIŽE $e \leq$ „malé“ **A** $e >$ „výrazně malé“ **PAK** Y je „velmi zhruba velké“

JESTLIŽE $e \leq$ „zhruba velké“ **A** $e >$ „malé“ **PAK** Y je „výrazně velké“

JESTLIŽE $e >$ „velmi zhruba velké“ **PAK** Y je „velmi zhruba velké“

Jelikož v tuto chvíli neznám fyzikální model systému a potřebuji, aby výsledný fuzzy regulátor pracoval co nejvíce optimálně, musím zvolit řádně jazykový popis jednoduchých fuzzy výrazů.

6.4.3 Stanovení souboru fuzzy výrazů

Pro jednotlivé fuzzy výrazy vyjadřující odchylku teploty e si empiricky určím hodnoty, které budou odpovídat danému výrazu. A to tak, že výraz s nejmenším významem položí roven 0 % e_{\max} a výraz s nejvyšším významem položí roven 100 % e_{\max} .

$e_{\text{nulové}} = 0$ (tj. 0% e_{\max})

$e_{\text{velmi výrazně malé}} = 0,1$ (tj. 3% e_{\max})

$e_{\text{velmi značně malé}} = 0,2$ (tj. 6% e_{\max})

$e_{\text{značně malé}} = 0,3$ (tj. 9% e_{\max})

$e_{\text{malé}} = 0,6$ (tj. 18% e_{\max})

Domnívám se, že pro oblast mezi 18% - 83,3 % není nutno definovat žádnou další fuzzy odchylku, protože v tomto pásmu je podle mého názoru nutno topit pomocí konstantního topného výkonu a tím dosáhnout co nejdříve okolí žádané teploty. V tomto pásmu podle mého názoru nehrozí riziko vzniku přehřívání.

$e_{\text{velké}} = 2,5$ (tj. 83,3% e_{\max})

$e_{\text{velmi zhruba velké}} = 3$ (tj. 100% e_{\max})

6.4.4 Stanovení souboru fuzzy hodnot

Pro jednotlivé fuzzy výrazy vyjadřující topný výkon si taktéž empiricky určím hodnoty, které budou odpovídat danému výrazu, a to velmi podobně jako u odchylky e , tedy výraz s nejnižším významem „velmi výrazně malý“ bude 0% Y_{\max} a výraz s nejvyšším možným významem „velmi velký“ bude 100 % Y_{\max} .

$Y_{\text{nulový}} = 0$

$Y_{\text{malý}} = 4$

$Y_{\text{zhruba malý}} = 5$

Y velmi zhruba malý = 6

Y velmi zhruba velký = 7

Y výrazně velký = 8

Y velmi velký = 9

Jelikož se tato závislost zdá být na první pohled lineární, jedná se s velkou pravděpodobností o fuzzy mutaci P (proporcionálního) regulátoru. Tento má výhodu v tom, že jeho rozhodovací schopnost je mnohem flexibilnější, protože pracuje pouze „téměř“ lineárně a jeho akční zásahy jsou blíže ideálním požadovaným hodnotám.

Na základě výše uvedených faktů jsem tedy navrhnul tuto funkci v jazyce C, která bude realizovat rozhodování o velikosti topného výkonu. Je velmi prostá, jen kopíruje stanovený fuzzy vztah.

6.4.5 Základní struktura programu regulátoru

Hlavní struktura programu by se dala rozdělit na části, a to podle úlohy, kterou má daná součást zastat:

- Hlavní cyklus regulátoru (tělo main),
- vlákno pro regulátor a pro čidla teploty,
- vlákno pro telnet server,
- vlákno pro gsm server.

6.4.6 Hlavní funkce programu regulátoru

```
int GetTopiciVykon(double zadanatep, double aktualtep)
{
    int topvyk = 0;
    if (aktualtep > zadanatep) return 0;
    if (zadanatep == aktualtep)
    {
        return 0;
    }

    if (aktualtep < zadanatep)
    {
        if (zadanatep - aktualtep >= 3) return 9;
        if (zadanatep - aktualtep >= 2.5) return 8;
        if (zadanatep - aktualtep >= 0.6) return 7;
        if (zadanatep - aktualtep >= 0.3) return 6;
        if (zadanatep - aktualtep >= 0.2) return 5;
        if (zadanatep - aktualtep >= 0.1) return 4;
    }
    return topvyk;
}
```

```
}
```

Tato funkce má pro názornost nahrazeny fuzzy názvy konstant čísly. Jinak by se místo těchto čísel v kódu objevily pomocí #define pojmenované konstanty.

6.4.7 Objekt pro práci se sériovým portem

Pro práci se sériovým portem slouží mnou psaný objekt Serial, jehož definice se nachází v souboru Serial.h, implementace v Serial.cpp.

Inicializace se provádí v konstruktoru nebo metodou Open, parametry jsou řetězec identifikující COM port, tedy „COM1“, dále pointer na DCB strukturu a nepovinně pointer na COMMTIMEOUTS strukturu:

```
void CSerial::CSerial(char*port, DCB*dcb, COMMTIMEOUTS* timeouts=NULL)
```

```
bool CSerial::Open(char*port, DCB*dcb, COMMTIMEOUTS* timeouts = NULL)
```

Zápis a čtení se provádí pomocí metod Write a Read, přičemž buffer je pointer na inicializovanou paměť o délce towrite, potažmo toread. Návrátová hodnota znamená počet zapsaných, potažmo načtených dat. Při návratu -1 došlo k chybě.

```
int CSerial::SyncWrite(BYTE * buffer, int towrite)
```

```
int CSerial::SyncRead(BYTE * buffer, int toread)
```

Ukončení se provádí metodou, bez parametrů:

```
bool CSerial::Close()
```

Ostatní metody zmiňovat není potřeba, jejich význam vstupní a výstupní hodnoty jsou popsány v komentářích.

Zdrojový kód v C++ je uveden v projektu VS 6.0 na přiloženém CD.

6.4.8 Objekt pro práci s paralelním portem

Pro práci s paralelním portem slouží psaný objekt ParalelPort, jehož definice se nachází v souboru ParalelPort.h, implementace v ParalelPort.cpp.

Neboť se v operačním systému Windows ® explicitně očekává využití vyššího programovacího rozhraní pro tisk, nejsou v API připraveny funkce pro přímou práci s rozhraním paralelního portu. Řešením je napsat si vlastní driver pro paralelní port, je to však dosti pracné.

Naštěstí Windows 95 umožňují přímý přístup na paralelní port pomocí assembleru – tedy strojového jazyka počítače nebo pomocí freeware řešení – „inpout32.dll“. Já jsem si vyzkoušel oba dva způsoby, a tak jsem ve svém kódu použil #define bloky, podle toho, zda-li chci nebo nechci použít assembler či „inpout32.dll“.

Pro přístup na paralelní port se používají fyzické porty systému – a to:

- řídicí registr port č. 0x378,
- datový registr port č. 0x379.

Princip využití těchto portů je následující:

- chci-li za zapisovat byte na paralelní port, zapíši jej do řídicího registru 0x378. Jakmile je byte zapsán, nastaví se datový registr 0x379 na zapsanou hodnotu,
- chci-li číst z paralelního portu, načtu postupně datový registr a poté řídicí registr. Po vyčtení byte z řídicího registru je v této návratové hodnotě uložená hodnota paralelního portu.

Při použití assembleru se pro zápis použije tato rutina:

```
1      _asm
2      {
3          push dx
4          push ax
5          mov dx, 0x378
6          mov al, value
7          out dx, al
8          xor ax, ax
9          in al, dx
10         mov retval, al
11         pop ax
12         pop dx
13     }
14     // návrat je v „retval“
```

Při použití assembleru se pro čtení použije tato rutina:

```
1      _asm
2      {
3          push dx
4          push ax
5          mov dx, 0x379
6          in al, dx
7          mov dx, 0x378
8          in al, dx
9          mov retval, al
10         pop ax
11         pop dx
12     }
13     // návrat je v „retval“
```

Používání „inpout32.dll“ je o něco složitější, je nutno nejprve zavolat načtení funkcí z „inpout32.dll“ pomocí funkce LoadLibrary, poté naplnit pointery na funkce (dle šablony v header souboru) a tyto funkce potom volat. Při ukončení programu se musí knihovna „inpout32.dll“ od programu odpojit – funkce FreeLibrary. Samotné volání vstupně výstupních funkcí je však jednodušší:

- Zápis dat: `this->Output(0x378, hodnota),`
- Čtení dat: `hodnota = this->Input(0x379).`

Zdrojový kód v C++ je uveden v projektu VS 6.0 na přiloženém CD.

6.5 Regulátor - součást GSM Server

GSM Server je součástí programu Regulátor. Sloučení funkcionality Regulátoru a GSM Serveru bylo velmi vhodné, neboť spolu obě dvě součásti sdílejí prostředky i paměť a jejich společná existence je tímto způsobem flexibilnější a všestranně výhodnější.

Tato součást se spouští při startu aplikace regulátoru v novém vlákne. Při spouštění se ověří spojení s mobilním telefonem a poté se vlákno dostane do hlavní smyčky. V této smyčce se cyklicky kontrolují SMS zprávy (dále jen „zprávy“) v mobilním telefonu. Při nalezení zprávy je tato zpracována a její obsah porovnán se seznamem podporovaných příkazů. Pokud zpráva syntakticky odpovídá podporovanému příkazu, je tento vykonán, vytvořena potvrzující zpráva a zpracovaná zpráva je z telefonu smazána. V ostatních případech je zpráva ignorována. Po ukončení fáze zpracování zpráv program čeká stanovenou dobu a poté se opět koloběh opakuje.

Při ukončení aplikace regulátoru je tato součást, tedy GSM Server, ukončena spolu s ním. Pro názornost je blokové schéma GSM serveru umístěno v příloze P IX a seznam použitelných příkazů se nachází v příloze P IX. Zdrojový kód v C++ je uveden v projektu VS 6.0 na přiloženém CD.

6.6 Regulátor - součást Telnet Server

Telnet Server je součástí programu Regulátor. Sloučení funkcionality Regulátoru a Telnet Serveru bylo opět, stejně jako u GSM Serveru, velmi vhodné, neboť spolu obě dvě součásti sdílejí, jak už bylo uvedeno dříve, prostředky i paměť.

Tato součást se spouští také při startu aplikace regulátoru v novém vlákně. Při spuštění se provede propojení („bind“) na lokální TCP/IP adresu a zahájí se hlavní smyčka programu serveru. V této probíhá TCP/IP naslouchání („listen“) na zvoleném portu, tj. 6000. Při spojení se provede převzetí spojení („accept“) a započne načítání dat z TCP/IP pípy. Při obdržení CR-LF (ASCII znaky 13 a 10) nebo při dosažení maximálního počtu příchozích znaků je provedeno zpracování příkazu. Pokud je tento příkaz podporován a jeho syntaxe je v pořádku, příkaz je proveden a na otevřené spojení se provede zápis dat obsahujících

výsledek příkazu. Jinak je tento příkaz ignorován a běh programu se vrací opět na fázi („listen“). Při ukončení aplikace regulátoru je tato součást, tedy Telnet Server, ukončena spolu s ním.

Pro názornost je blokové schéma Telnet serveru umístěno v příloze P X. Seznam použitelných příkazů se nachází v příloze P X. Zdrojový kód v C++ je uveden v projektu VS 6.0 na přiloženém CD.

6.7 Aktivní serverový skript PHP

Tyto skripty, jak již bylo naznačeno, budou vytvořeny ve skriptovacím jazyce PHP. Tento se jeví jako nejvhodnější kandidát pro svou jednoduchost a účelnost. Důležitý je samozřejmě také ten fakt, že interpreter jazyka PHP je k dispozici zdarma. [39]

Pro zjednodušení je pro rozhraní WWW serveru použit pouze jeden skript. Tento se pomocí ovládacích prvků v jazyce HTML odkazuje sám na sebe a v záhlaví tohoto skriptu se nachází samotný výkonný kód PHP, který již vykonává potřebné funkce. Parametry se předávají pomocí HTTP metody POST a GET, jsou tedy předávány v HEADERECH požadavku („requestu“).

Jak získá PHP skript informaci o tom co se děje v regulátoru a jak bude moci regulátor nastavovat? Je to prosté. PHP jako takové ve své základní verzi podporují socketovou komunikaci, tedy komunikaci po síti pomocí protokolu TCP/IP. Proto bude využívat služeb Telnet serveru zabudovaného v regulátoru. Výsledky budou interpretovány jako vizuální data přímo v HTML kódu. Zdrojový kód PHP skriptu je uveden v souboru index.php na přiloženém CD.

7 REGULOVANÁ SOUSTAVA

7.1 Výroba plošného spoje foto cestou

Prvním krokem pro vytvoření plošného spoje je jeho návrh. Na základě rady vedoucího diplomové práce jsem pro návrh použil demoverzi elektrotechnického programu Eagle v4.11.

Eagle v4.11 je velmi vhodný pro návrh plošných spojů, neboť jeho obsahuje databázi více než 1000 existujících součástí včetně jejich rozměrů a typů pouzder, které se pro návrh dají použít. Program Eagle má propracované grafické rozhraní a intuitivní ovládání myší i klávesnicí. Má také tiskové sestavy připravené pro výrobu plošných spojů foto-citlivou cestou.

Návrh plošného spoje je v demoverzi možno umístit pouze na desku o maximálních rozměrech 8 x 12 cm a maximální počet použitých součástí nebo jejich vzájemných spojů je 100.

Druhým krokem je výběr vhodného materiálu. Tímto lze bezesporu prohlásit mědi pokrytý cuprexit s připravenou foto-citlivou vrstvou.

Třetím krokem je výroba plošného spoje. Tato se dá rozdělit do níže uvedených fází:

- Vytištění vzhledu tištěného spoje v měřítku 1:1 na průhlednou folii,
- Osvícení cuprexitové desky zdrojem UV záření (např. rtuťová výbojka) přes potišťnou průhlednou folii po potřebnou dobu (tuto je nutno vyzkoušet pokusem),
- Vyvolání osvícených spojů ve vodném roztoku 1,5 % NaOH (Hydroxid sodný) při teplotě roztoku 15 – 20 °C,
- Odleptání přebytečné mědi v Fe₂O₃ (Oxid železitý) – pomalejší reakce, nebo vodném roztoku 30% HCl (Kyselina chlorovodíková) a 100% H₂O₂ (Peroxid vodíku) v objemovém poměru 1 : 1 : 1 – rychlejší chemická reakce – navíc silně exotermická,
- Vrtání pájecích otvorů mikrovrtáčkou – 0,8 mm pro většinu součástek nebo 1,2 mm pro větší průměry konektorů,
- Osazení a zapájení součástek podle schématu. [37]

7.2 Výroba Relay Desk v 1.0

Výrobní proces jsem začal návrhem vhodného schématu. Schéma jsem vytvořil v demoverzi programu určeného pro návrh schémat a tištěných spojů – Eagle v4.11. Demoverze byla omezena velikostí výsledného plošného spoje a počtem spojů, což mě ale příliš netrápilo, neboť reléová deska je v tomto mém konkrétním případě jednoduchá a omezení mě tudíž netrápilo.

7.2.1 Potřebné součástky

- 2 x relé Schrack 210012,
- 2 x odpor 1k5,
- 2 x odpor 2k2,
- 2 x odpor 1k2,
- 2 x obyčejná dioda,
- 2 x LED,
- 2 x kondenzátor M1,
- 2 x tranzistor KSY34D1,
- 2 x dvojité šroubovací svorky (rozteč 5mm),
- 1 x konektor napájecí 4 pinový - samec (rozteč pinů 5 mm).

Pro názornost přikládám schéma a vzhled plošného spoje v příloze P XI.

Výrobní proces jsem zahájil nákupem potřebných součástek. Pokračoval jsem výrobou plošného spoje – desku pro plošný spoj jsem zvolil z poměděného cuprexitu. Plošný spoj jsem vyrobil fotocestou. Následovalo vrtání pájecích otvorů, osazení součástkami a nakonec pájení.

Celý výrobní proces tištěného spoje reléové desky mě zabral jeden pracovní den. (o výrobě fotocitlivou více v podkapitole 7.1).

7.3 Výroba Com Switch v1.0

Výroba přepínače sériového portu byla naprosto identická s výrobou reléové desky - používal jsem stejný nástroj na navržení schématu i desky a také jsem postupoval stejně, když jsem tento plošný spoj vyráběl.

7.3.1 Potřebné součástky

- 3 x integrovaný obvod 74HCT138,
- 1 x integrovaný obvod 74153,
- 1 x integrovaný obvod 7406,
- 1 x integrovaný obvod 7408,
- 6 x blokovací kondenzátor M1,
- 3 x odpor 10k,
- 3 x odpor 2,7k
- 3 x LED,
- 3 x konektor CANNON9 samec– provedení pro plošný spoj,
- 1 x konektor CANNON9 samec – provedení pro plošný spoj,
- 2 x dvojité šroubovací svorky (rozteč 5mm),
- 1 x napájecí 4 pinový konektor – samec (rozteč pinů 5 mm).

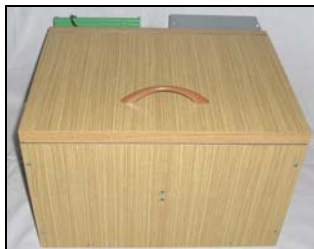
Schéma a vzhled plošného spoje se nacházejí v příloze P XII.

Protože se mi nepodařilo vytvořit ideální návrh rozložení součástek na plošném spoji, bylo nutné v některých případech provádět přemostění spojů „vzduchem“. K přemostění jsem použil tenký měděný drát.

Celý výrobní proces tištěného spoje přepínače COM portů mě zabral dva pracovní dny, počítaje výrobu schématu, nákup součástí, příprava plošného spoje, vrtání, osazení a zapájení. (o výrobě fotocitlivou více v podkapitole 7.1)

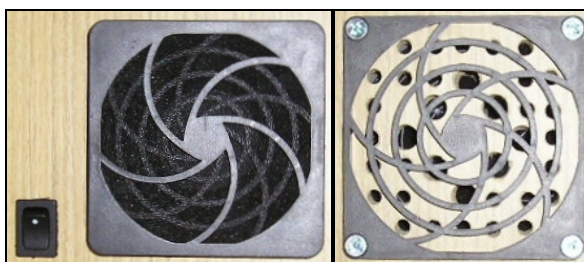
7.4 Sestavení korpusu simulátoru regulované soustavy

Jako korpus byla vybrána dřevotřísková krabice. Je poskládána z hlavní části a víka. Víko je ke hlavní části připevněno dvěma nábytkářskými panty a je vybaveno držátkem, pro jednodušší otevírání. Rozměry bedny jsou (š x v x d) 40 cm x 25,3 cm x 30,4 cm.



Obr. č. 24 – Pohled na krabici shora

V levém boku krabice se nachází otvor s ventilátorem a v pravém boku se nachází mřížka ventilačních děr. Tyto součásti jsou implementovány v krabici proto, aby bylo možno simulovat tepelné ztráty.



Obr. č. 25 - Vlevo pohled na ventilátor se spínačem, vpravo pohled na odkrytované ventilační díry

Důležité spínače, které se nacházejí na vnější straně krabice, jsou zevnitř zakrytovány plastovými krytkami a tím je veškerá kabeláž, která k nim vede řádně schována.

Krabici jsem si sestavil z předem nařezaných desek. K upevnění jednotlivých částí jsem použil speciální šrouby na dřevotřísku, tzv. konfirmáty. Po sestavení hlavní části krabice jsem hrany krabice odýhoval a provedl montáž pantů. K pantům jsem následně namontoval víko a pak k víku držák. Dále jsem pomocí vrtačky a hrubého pilníku vytvořil otvory pro všechny spínače a pro ventilátor. Naposledy jsem vytvořil symetrický grid (mřížka) 37 ventilačních děr na protější straně od ventilátoru.



Obr. č. 26 - Pohled na otevřenou krabici shora

Při vytváření elektroinstalace jsem nejprve do krabice umístil průchozí plastové lišty, a to tak, že jsem si vždy připravil lištu o potřebné délce a tuto jsem pak vzápětí přišrouboval potřebným počtem samopřezných šroubů. Potom jsem přidal objímku žárovky a poté jsem provedl instalaci kabeláže, s tím, že jsem si připravil průchozí otvory. Na konci této fáze jsem propojil všechny spínače.

Při instalaci jednotlivých zařízení jsem postupoval takto: nejprve jsem upevnil reléovou desku, poté jsem připravil umístění pro dodatečnou svorkovnici, poté jsem připevňoval přepínač sériového portu a naposledy jsem umístil Biscuit PC.

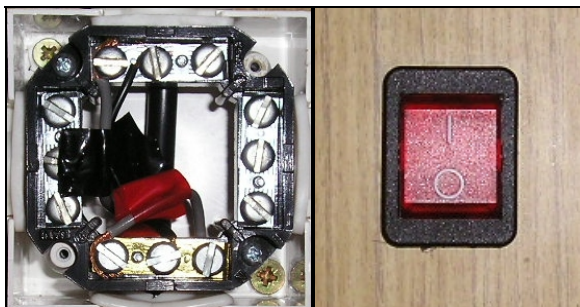
7.5 Instalace rozvodů elektrické energie

V krabici se nachází rozvod 230V (nízké napětí: 50V až 300V). Dále se v krabici nachází rozvody napětí stejnosměrného o velikosti 12V (malé napětí: 0V – 50V). Tyto jednotlivé rozvody jsou od sebe odděleny bezpečnou vzdáleností, a nikde se nestane, že by se rozvody křížily nebo snad byly umístěny souběžně v jedné liště.

Hlavní spínač se nachází v levém boku krabice. Obsahuje v sobě světelnou indikaci. Pokud je sepnutý a prochází jím proud, pak svítí.

7.5.1 Instalace rozvodů 230V

Na tento rozvod je napojen hlavní spínač a na něj je napojena svorkovnice, která se nachází v zadní části krabice. Z této svorkovnice je napájen zdroj stejnosměrného napětí a zdroj tepla (tedy žárovka).



Obr. č. 27 – Pohled na svorkovnici (vlevo)
a na síťový vypínač (vpravo)

Svorkovnice velkého napětí jsem zakrytoval a zejména jsem dbal, aby se nikde neobjevil ani náznak špatně izolovaného drátu, kterým by teklo velké napětí.

7.5.2 Instalace rozvodů nízkého napětí +5V a +12V

Tento rozvod napájí embedded PC, ventilátor, reléovou desku a přepínač sériového portu. Protože ze zdroje nevede dostatečný počet napájecích kabelů, je na zadní stranu krabice přidána jedna dodatečná svorkovnice, ze které se dá natáhnout vedení jak +5V tak +12V voltů stejnosměrného napětí.

7.5.3 Kabeláž

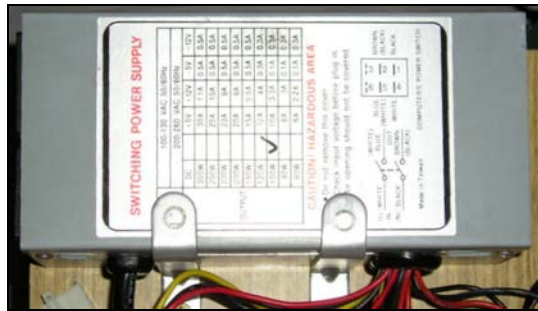
Jednotlivé svazky kabelů, které slouží k vedení elektrické energie jsou vyrobeny z mědi a izolovány. Kabely s velkým napětím ctí standardní označení, tj. hnědý je fáze, modrý je střední pracovní proud a zelený je zem. Průměr mědi ve nízkonapěťových kabelech je 2 mm. Kabely, ve kterých teče malé napětí jsou označeny červeným pruhem, ty které protahují zem jsou celé černé. Průměr mědi v malonapěťových kabelech je 1 mm. Kabeláž pro nízké napětí je kompletně vedena v plastových lištách. Kabeláž pro malé napětí je vedena částečně v lištách, částečně vzduchem.

7.6 Elektrické spotřebiče

Mezi elektrické součásti krabice patří zdroj napětí, topné těleso (žárovka), ventilátor, reléová deska a přepínač sériových portů a embedded PC.

7.6.1 Montáž zdroje stejnosměrného napětí

Jedná se o počítačový zdroj o výkonu 100 W, který převádí střídavé napětí o velikosti 230V na stejnosměrné napětí +5V a +12V. Zdroj má vyvedeny kabely se standardními koncovkami, celkem 4 svazky kabelů pro napájení periférií. Kabely jednotlivých svazků jsou barevně odděleny, a to takto: žlutý obsahuje +12V, červený +5V, černý je zem.



Obr. č. 28 – Zdroj stejnosměrného napětí

7.6.2 Instalace topného tělesa

Žárovka je napájena střídavým napětím o velikosti 230 V. Kabeláž k ní vedoucí je tažena z hlavní svorkovnice. Tento elektrický obvod je přerušen přes první relé. Příkon žárovky je určen na 60 W.



Obr. č. 29 – Žárovka v objímce

7.6.3 Montáž ventilátoru

Ventilátor je napájen stejnosměrným napětím o velikosti +12V. Jedná se standardní počítačový ventilátor o průměru 8 cm. Tento má výkon 2,6 W. Pro tento ventilátor existují dva režimy. Jeden je automatický, a druhý je manuální. Režim se přepíná pomocí přepínače na zadní straně krabice. Pokud je ventilátor v režimu automatickém, pak spínač umístěný vlevo od ventilátoru nemá žádný význam a ventilátor je spouštěn pomocí druhého relé,

pokud se však ventilátor přepne do manuálního režimu, pak se pomocí spínače vlevo od něj ventilátor spíná a vypíná. O tom, který režim je zrovna aktivní je možné se dozvědět podle stavu indikačních LED, které se nacházejí po levé a pravé straně přepínače režimů.



Obr. č. 30 – Pohled na ventilátor zevnitř



Obr. č. 31 – Pohled na spínač ventilátoru



Obr. č. 32 – Pohled na přepínač režimů ventilátoru

7.6.4 Montáž reléové desky (Relay Desk v 1.0)

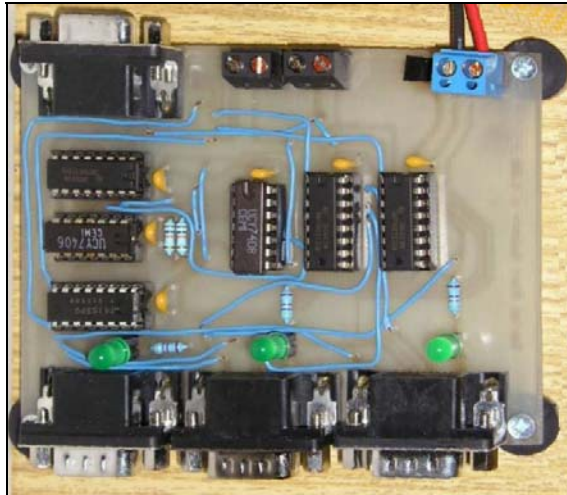
Je napájena stejnosměrným napětím o velikosti +5 V. Je umístěna v zadní části krabice a je k ní připevněna 4 průchozími šrouby o průměru 2 mm. Od krabice je oddělena pomocí 0,4 cm vysokých plastových podložek.



Obr. č. 33 – Pohled na Relay Desk v1.0

7.6.5 Montáž přepínače sériových portů (Com Switch v1.0)

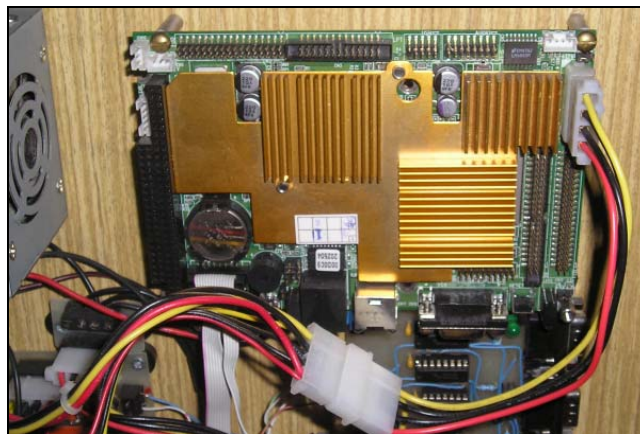
Tento je napájen také stejnosměrným napětím o velikosti +5 V. Je umístěn v zadní části krabice poblíž reléové desky. Ke krabici je připevněn průchozími šrouby o průměru 2 mm. Od krabice je oddělen pomocí 0,4 cm vysokých plastových podložek.



Obr. č. 34 – Pohled na Com Switch v 1.0

7.6.6 Instalace Biscuit PC Advantech® PCM-5822

Napájecí napětí tohoto zařízení je +5V. Je umístěno v zadní části krabice a je uchyceno k bedně pomocí univerzální plechové lišty, a tato je připojena ke krabici pomocí 2 průchozích šroubů o průměru 2 mm. (více o Biscuit PC viz kapitola 5.1) Způsob přichycení Biscuit PC ke krabici je znázorněn na následujícím obrázku:



Obr. č. 35 – Pohled z boku a shora na uchycené Biscuit PC Advantech® PCM-5822

8 POPIS STARTU, INICIALIZACE A PRACOVNÍHO CYKLU MONITOROVACÍHO A ŘÍDICÍHO SYSTÉMU

Tato kapitola pojednává o činnosti monitorovacího a řídicího systému. Činnost se dá rozdělit do čtyř základních fází, a to jsou:

- Spuštění hardware,
- Inicializace hardware a operačního systému,
- Inicializace software regulátoru a jeho pracovní cyklus,
- Eventuelní ukončení.

8.1 Spuštění hardware

Ke spuštění hardware dojde při zahájení dodávky elektrické energie do počítačového zdroje. Při spouštění hardware dochází nejprve k zavádění operačního systému. Tento se zavádí pomocí zavaděče (což je program, který je vyvolán ihned při spuštění hardware). Tento zavaděč se u Windows 95 nachází v boot sektoru disku (boot sektor disku je 1-ní sektor na 0-tém povrchu, 0-tém cylindru a 0-té stopě). Zavaděč zavolá jádro systému – v tomto případě io.sys, který již provede samotné spuštění operačního systému.

Při svém spuštění provede kontrolu spouštěcí oblasti operačního systému, zkontroluje, zda-li je dostatek volné operační paměti, zavede potřebné softwarové ovladače a spustí grafické rozhraní operačního systému.

8.2 Inicializace software regulátoru

Probíhá po zavedení programu do paměti. Nejprve je otestována přítomnost podpůrného hardware. Pokud vše probíhá dobře, tak se na pár vteřin rozsvítí žárovka a sepne ventilátor a poté jsou v cyklu rozsvíceny zelené LED na přepínači sériového portu. Podle odezvy od paralelního portu se zkontroluje, zda-li tato fáze proběhla úspěšně. Jako poslední se provede test přítomnosti mobilního telefonu v systému. Není-li mobilní telefon nalezen, pak se nastaví příznak oznamující programu, že se nemá pokoušet s tímto zařízením pracovat.

Opětná detekce se provádí za deset minut až dokud není zařízení nalezeno nebo do ukončení programu. Tento příznak se také nastaví pokud komunikace s mobilním telefo-

nem s jakéhokoliv důvodu selže, aby se získal čas, který by eventuelně mobilní telefon potřeboval k obnovení své činnosti.

Pokud se inicializace mobilního telefonu povede, program automaticky spustí separátní vlákno, které začne obsluhovat mobilní telefon a které podle zadaných pravidel bude se zařízením komunikovat.

Jako další akce, která se provede, je vytvoření vlákna, které začne poslouchat na ethernet rozhraní na stanoveném portu. Při selhání této součásti je regulátor nemilosrdně ukončen a je vytvořen informační soubor obsahující okolnosti neúspěchu inicializace. Jako poslední akce, která se provede, bude vytvoření vlákna, které začne načítat z připojených čidel teplotu. Pokud se z jakéhokoliv důvodu nepovede inicializovat toto vlákno, regulátor je bez milosti shozen a je opět vytvořen informační soubor s popisem chyby.

8.3 Pracovní cyklus

Po úspěšné inicializaci všech potřebných součástí je program připraven pracovat. Vstoupí do hlavního cyklu programu ve kterém ve smyčce provádí stále dokola tuto činnost:

- Ověří nové úkoly, které mohly být systému zadány pomocí TCP/IP serveru nebo mobilního telefonu. Pokud úkoly najde, připraví jejich provedení.
- Načte venkovní a vnitřní teplotu systému.
- Vloží načtené teploty do rozhodovací fuzzy funkce, která spočítá topný výkon, dle potřeby nastaví nový topný výkon.
- Pokud přijde čas ke zpracování sms, umožní vláknu mobilního telefonu načíst sms. Během této doby se preventivně zastavuje topení a čeká se po maximální povolenou dobu na dokončení tohoto vlákna. Pokud vlákno neskončí včas, je restartováno.
- Na konci cyklu ověří, zda-li nebyl doručen příkaz o ukončení programu - pokud ano, slušně ukončí všechny spuštěná vlákna a ukončí se.

9 FÁZE VZNIKU PROJEKTU A ROZPOČET

Při vytváření návrhu celého řídicího a monitorovacího systému bylo nutno zvážit také časové a finanční hledisko. Toto sice v tomto konkrétním případě přílišnou roli nehrálo, neboť není součástí relevantního obchodního projektu, ovšem v případech, kdy se jedná o vývoj systému, který má být obchodně využit, je nutno zvážit náročnost, se kterou bude tento systém vyroben. Bylo proto dobré, označit si alespoň čtyři základní fáze, které se týkají vzniku takového systému, a to jsou dle mého názoru a praktických zkušeností tyto:

9.1 Vytváření návrhu

Tato fáze je nejvíce náročná na duševní práci návrháře. Při vytváření návrhu se musejí brát na potaz všechny aspekty týkající se celého systému. Musí se počítat s každou eventualitou a připravit všechny podklady tak, aby se v průběhu realizace samotný návrh buď vůbec neměnil nebo měnil velmi málo. Bývá často úkolem pro vývojový tým, tedy tým analytiků a odborníků na vývoj, protože jeden samotný člověk může při narůstajícím objemu vznikajících materiálů velmi lehce udělat chybu, a ta se pak může vymstít během fáze realizace. Tato fáze může trvat zhruba 1/3 celkové doby určené pro vývoj až po realizaci prototypu.

Bohužel u tohoto výše uvedeného systému nebylo z časového hlediska možné vzít na potaz úplně všechny možné úhly pohledu a tím pádem se připravit na všechny možné události a situace, ovšem snažil jsem se, aby se podle výše uvedeného návrhu dal takový systém vytvořit, a to s co pokud možno nejmenšími změnami během realizační fáze. Mě tato fáze zabrala celou jednu polovinu doby, kterou jsem měl vyhrazenou na návrh a realizaci tohoto systému. Ovšem s tím, že jsem ji spojil s fází druhou. Nemohu tedy ani zhruba určit jak dlouho v mém případě trvaly jednotlivé fáze.

9.2 Kontrola návrhu

Ve fázi kontroly návrhu je nutno předat veškerou vytvořenou dokumentaci pokud možno jinému týmu, anebo v případě stejného týmu lidí nechat jistý časový interval mezi dokončením dokumentace a její následnou kontrolou. Tým bývá složen opět z analytiků či konzultantů. Tato fáze se často spojuje s fází první. Může trvat zhruba 1/6 celkové doby určené pro vývoj a realizaci daného projektu.

Já jsem byl nucen tuto fázi realizovat zvláště, neboť nebyl nikdo, kdo by mi mohl opravit mé vlastní návrhy. Přesto očekávám, že jsem touto fází kontroly návrhu prošel bez problémů.

9.3 Realizace

Fáze realizace je celkem náročná část. Zde zkontrolovanou dokumentaci přebírají lidé, kteří jsou schopni dané zadání realizovat, tedy technici a programátoři. V této fázi se již začínají hledat dodavatelé pro navržené součásti systému. Testují se navržené součásti a jejich chování v praxi. Také se sleduje, jak reaguje dodavatel na vzniklé obtíže plynoucí z použití jeho výrobku. V této fázi je klíčové dostat projekt do takového stavu, aby byly na konci této fáze známy všechny potíže plynoucí z návrhu a jejich řešení a aby se dalo co nejdříve připravené zařízení nebo systém testovat. Tato fáze v reálných podmínkách trvá zhruba 2/6 celkového času přiděleného na daný projekt.

Moje zkušenosti s touto fází jsou spíše záporné. Neboť při fázi realizace jsem opravdu narazil na překážky, které vyplynuly z navrženého řešení. Vždy se mi však podařilo najít vhodné řešení, které umožnilo dodržet postupy uvedené v návrhu. Doba strávená v této fázi odpovídala očekávaným dvěma šestinám celkové doby.

9.4 Testování

Poslední fází je testování již připraveného konečného řešení. V této fázi se provádějí zátěžové testy, které odhalují poslední možné chyby, které se dají zjistit do konce vývoje. Paradoxní je, že v této fázi se odhalí stejné množství chyb jako při fázi realizační, ovšem odstup mezi jednotlivými chybami bývá delší a jejich řešení bývá jednodušší. V této fázi se dá hovořit již o před-produkční verzi zařízení/systému. Je důležité, aby zařízení/systém testoval někdo, kdo nebyl současně v jeho vývoji nebo při realizaci, protože lidé zainteresovaní ve vývoji či realizaci bývají zatíženi předpoklady, že některé součásti systému „prostě chybu obsahovat nemohou“ a zpomaluje se ladění chyb.

Já osobně jsem fázi testování nechal na svých spolupracovnících, kteří dostali základní popis chování mnou navrženého systému, včetně způsobu, jak s ním komunikovat. Tím, že se snažili se systémem pracovat, nacházeli chyby, které jsem sám při realizační části najít nenašel.

9.5 Rozpočet

Rozpočet je část velmi citlivá, protože na jeho celkové výši záleží, co všechno se u daného zařízení nebo systému může realizovat. Jeho výše bývá často důvodem ke dlouhým debatám a někdy k hádkám. Přesto jeho stanovení je klíčové pro celý projekt. Bývá stanoven na jednotlivé fáze a dále na jednotlivé součásti systému a bývá vytvořen fundovaným ekonomickým pracovníkem, kterému dodává podklady pracovník, který má na starosti vypracování návrhu.

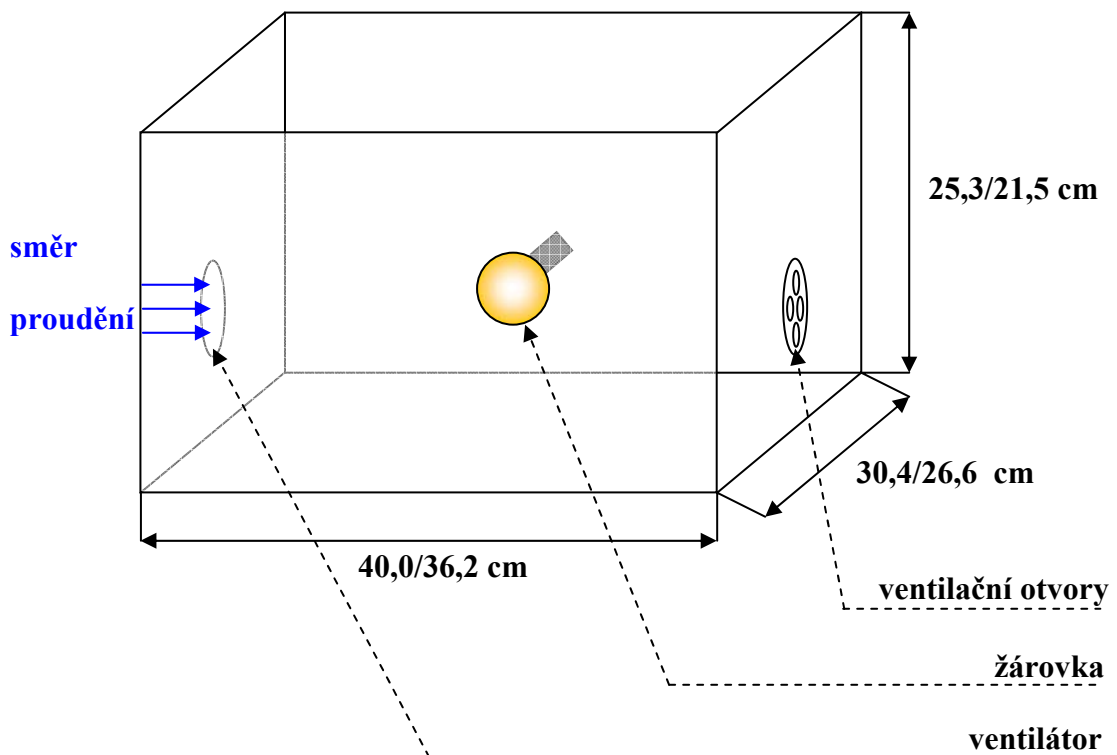
10 OVĚŘENÍ ČINNOSTI ŘÍDICÍHO A MONITOROVACÍHO SYSTÉMU POMOCÍ REÁLNÉ REGULOVANÉ SOUSTAVY

10.1 Fyzikální základy modelu reálné tepelné soustavy

Navržená tepelná soustava jde popsat pomocí tepelně bilanční rovnice. Tato rovnice se vytváří na základě naměřených veličin. Část základních údajů o soustavě je již částečně uvedena v předchozích kapitolách, je to však málo, proto jsou požadované informace doplněny, aby bylo možno provádět stanovené výpočty.

10.1.1 Základní informace o tepelné soustavě

Jako první je nutné vytvořit si grafické znázornění tepelné soustavy, a vyznačit si v něm všechny potřebné údaje.



Obr. č. 36 – Grafické znázornění reálné soustavy

Za další je potřeba si vyjádřit potřebné veličiny spojené s touto soustavou:

- **Rozměry:**

- Vnější: (š x h x v)

$$40,0 \times 30,4 \times 25,3 \text{ [cm]}$$

- Vnitřní rozměry: (š x h x v)

$$36,2 \times 26,6 \times 21,5 \text{ [cm]}$$

- Tloušťka stěny: 1,9 cm

- **Plochy:**

- Celková plocha bedny vevnitř:

$$S_{\text{celkem_vevnitř}} = 2 \cdot (a \cdot b) + 2 \cdot (b \cdot c) + 2 \cdot (a \cdot c)$$

$$S_{\text{celkem_vevnitř}} = 2 \cdot (36,2 \cdot 26,6) + 2 \cdot (26,6 \cdot 21,5) + 2 \cdot (36,2 \cdot 21,5)$$

$$= 4626,34 \text{ cm}^2 = 0,463 \text{ m}^2$$

- Celková plocha bedny venku:

$$S_{\text{celkem_venku}} = 2 \cdot a \cdot b + 2 \cdot b \cdot c + 2 \cdot a \cdot c$$

$$S_{\text{celkem_venku}} = 2 \cdot (40 \cdot 30,4) + 2 \cdot (30,4 \cdot 25,3) + 2 \cdot (40,0 \cdot 25,3) = 5994,24 \text{ cm}^2$$

$$= 0,599 \text{ m}^2$$

- Plocha otvorů:

$$S_{\text{díra_ventilátor}}: r = 4 \text{ cm,}$$

(Plocha listů ventilátoru je 40 % z otvoru, tedy díra bude 60 %.)

$$S_{\text{díra_ventilátor}} = \Pi \cdot r^2 \cdot 0,6 = \Pi \cdot 4^2 \cdot 0,6 = 30,16 \text{ cm}^2$$

$$S_{\text{ventilační_otvory}} = S_{\text{díra_ventilátor}} \cdot \text{počet_děr}$$

$$1 \text{ díra cca } 0,45 \text{ cm}^2, \text{ počet děr} = 37,$$

$$S_{\text{ventilační_otvory}} = S_{\text{díry}} \cdot \text{počet_děr} = 0,45 \cdot 37 = 16,65 \text{ cm}^2$$

$$S_{\text{otvorů}} = S_{\text{díra_ventilátor}} + S_{\text{ventilační_otvory}} = 30,16 + 16,65 = 46,81 \text{ cm}^2$$

- Plocha bedny uvnitř bez děr:

$$S_{\text{uvnitř}} = S_{\text{celkem_vevnitř}} - S_{\text{otvorů}} = 4626,34 - 46,81 = 4579,53 \text{ cm}^2 = 0,458 \text{ m}^2$$

- **Objemy:**

- Objem bedny

$$V_{\text{bedna}} = a \cdot b \cdot c = 40,0 \cdot 30,4 \cdot 25,3 = 30764,8 \text{ cm}^3 = 0,031 \text{ m}^3$$

- Objem prostoru uvnitř celkem:

$$V_{\text{uvnitř_celkem}} = a \cdot b \cdot c$$

$$V_{\text{uvnitř_celkem}} = 36,2 \cdot 26,6 \cdot 21,5$$

$$V_{\text{uvnitř_celkem}} = 20702 \text{ cm}^3 = 0,021 \text{ m}^3$$

- Objem žárovky:

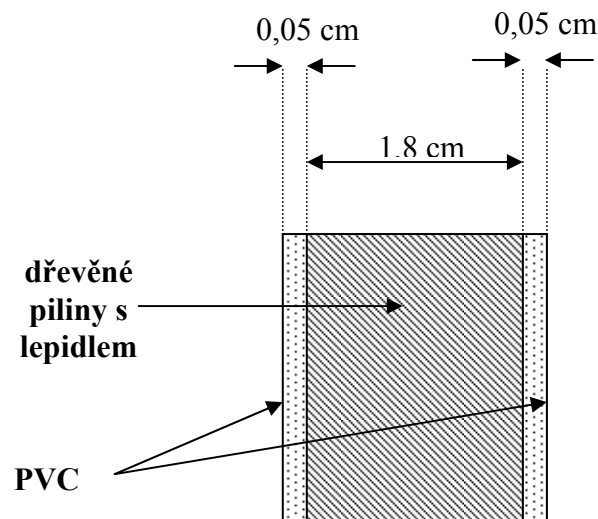
$$V_{\text{žárovky}} = 50 \text{ cm}^3 = 0,05 \text{ dm}^3 = 0,00005 \text{ m}^3$$

- Objem plynu uvnitř:

$$V_{\text{plyn_uvnitř}} = V_{\text{uvnitř_celkem}} - V_{\text{žárovky}} = 0,021 - 0,00005 = 0,02095 \text{ m}^3$$

- Objem stěn krabice:

$$V_{\text{stěny}} = V_{\text{bedna}} - V_{\text{uvnitř_celkem}} = 30764,8 - 20702 = 10062,80 \text{ cm}^3 = 0,01 \text{ m}^3$$



Obr. č. 37 – Zobrazení řezu dřevotřískou

- **Látkové složení**

- **Vzduch:**

Měrná tepelná kapacita:

$$c_p = 1000 \text{ J.kg}^{-1}.\text{K}^{-1}$$

Hustota:

$$\rho = 1,29 \text{ kg} \cdot \text{m}^{-3}$$

- **Dřevotříska:**

Složení:

Dřevo + PVC

Dřevo 98%, PVC 2 %

Měrná tepelná kapacita:

$$c_p = 1150 \text{ J.kg}^{-1}.\text{K}^{-1}$$

Hustota:

$$\rho = 628 \text{ kg} \cdot \text{m}^{-3}$$

10.1.2 Tepelně bilanční rovnice

Základní vztah, na kterém tato rovnice stojí, je zákon zachování energie. Z něho se dá vytvořit obecný vzorec pro vytvoření této tepelné bilance.

součet energií ve vstupních proudech + dodané teplo = součet energií ve výstupních proudech + vykonaná práce + akumulace energie v systému tedy

$$H + Q = H_v$$

Na základě této obecné formule a získaných informací o tepelné soustavě jsem vytvořil tuto bilanční rovnici:

$$P = V_{vzd} \cdot \rho_{vzd} \cdot c_{p_{vzd}} \cdot \frac{dt}{d\Gamma} + V_{bedny} \cdot \rho_{bedny} \cdot c_{p_{bedny}} \cdot \frac{dt}{d\Gamma} + K \cdot S_{bedny_uvnitř} \cdot (t_0 - t_1)$$

kde:

P topný výkon

$V_{\text{vzd}}, V_{\text{bedny}}$	objem vzduchu v bedně, objem hmoty bedny
$c_{p_{\text{vzd}}}, c_{p_{\text{bedny}}}$	měrná tepelná kapacita vzduchu, měrná tepelná kapacita hmoty bedny
$\frac{dt}{dT}$	derivace času podle teploty
K	součinitel přestupu tepla hmotou bedny
$S_{\text{bedny_uvnitř}}$	plocha vnitřního prostoru bedny, která pohlcuje vzniklé teplo
$(t_0 - t_1)$	rozdíl teplot, kde t_0 je teplota vně bedny a t_1 teplota uvnitř bedny

Nyní je nutno spočítat neznámou K (součinitel přestupu tepla bedny), a to pomocí vzorce:

$$K = \frac{1}{\frac{1}{\alpha_1} + \frac{\delta_1}{\lambda_1} + \frac{\delta_2}{\lambda_2} + \frac{\delta_3}{\lambda_3} + \frac{1}{\alpha_2}}$$

Kde:

$$\frac{1}{\alpha_1} \quad \text{převrácená hodnota koeficientu přestupu tepla vzduchu uvnitř bedny}$$

$$\frac{\delta_1}{\lambda_1} = \frac{\delta_3}{\lambda_3} \quad \text{podíl tloušťky vrstvy a součinitele teplotní vodivosti vrstvy-PVC}$$

$$\frac{\delta_2}{\lambda_2} \quad \text{podíl tloušťky vrstvy a součinitele teplotní vodivosti vrstvy-dřevotřísky}$$

$$\frac{1}{\alpha_2} \quad \text{převrácená hodnota koeficientu přestupu tepla vzduchu vně bedny}$$

$$\alpha_1 = c \cdot (Gr_1 \cdot Pr_1)^n$$

$$Gr_1 = \text{Grashofova kritériální rovnice pro teplo} = \frac{d^3 \cdot g \cdot \beta \cdot \Delta t}{\nu^2} \rightarrow \text{po dosazení:}$$

$$d^3 = 0,2683 = 0,0192 [\text{m}^3] \quad \text{- nejmenší rozměr bedny umocněný na }^3$$

$$g = 9,81 [\text{m} \cdot \text{s}^{-2}] \quad \text{- tíhové zrychlení}$$

$$\beta=1/T_s \text{ [K}^{-1}\text{]} \quad - \text{ součinitel objemové roztažnosti plynů, } T_s=299,15 \text{ K}$$

$$\Delta t=1 \text{ [K]} \quad - \text{ rozdíl teplot vně a uvnitř, zde nehraje roli,}$$

dosazeno = 1

$$\nu_{226^\circ\text{C}}=16,2 \cdot 10^{-6} \text{ [m}^2\text{s}^{-1}\text{]} \quad - \text{ kinematická viskozita vzduchu při teplotě } 26^\circ\text{C}$$

pro $Gr_1 = 107$ je určena hodnota $Pr_1 =$ Prandtlova kriteriální rovnice = 0,73

$Gr_1.Pr_1 = 7,3 \cdot 106$, z tabulek se dosadí hodnoty c a n , a ty budou $c=0,54$ a $n=1/4$

$$\alpha_1 = 28$$

$$Gr_2 = \text{Grashofova kriteriální rovnice pro teplo} = \frac{d^3 \cdot g \cdot \beta \cdot \Delta t}{\nu_{t^\circ\text{C}}^2} \rightarrow \text{po dosazení:}$$

$$d^3=0,2683=0,0192[\text{m}^3] \quad - \text{ nejmenší rozměr bedny umocněný na 3}$$

$$g = 9,81 \text{ [m.s}^{-2}\text{]} \quad - \text{ tíhové zrychlení}$$

$$\beta=1/T_s \text{ [K}^{-1}\text{]} \quad - \text{ součinitel objemové roztažnosti plynů, } T_s=299,15 \text{ K}$$

$$\Delta t=1 \text{ [K]} \quad - \text{ rozdíl teplot vně a uvnitř, zde nehraje roli,}$$

dosazeno = 1

$$\nu_{226^\circ\text{C}}=16,2 \cdot 10^{-6} \text{ [m}^2\text{s}^{-1}\text{]} \quad - \text{ kinematická viskozita vzduchu při teplotě } 26^\circ\text{C}$$

pro $Gr_2 = 107$ je určena hodnota $Pr_2 =$ Prandtlova kriteriální rovnice = 0,73 (strana 3)

$Gr_2.Pr_2 = 7,3 \cdot 106$, z tabulek se dosadí hodnoty c a n , a ty budou $c=0,54$ a $n=1/4$

$$\alpha_2 = 28$$

$$\delta_1 = \delta_3 = 0,5 \times 10^{-3} \text{ [m]}$$

$$\delta_2 = 1,8 \times 10^{-2} \text{ [m]}$$

$$\lambda_1 = \lambda_3 = 0,15 \text{ [W.m}^{-1}\text{.K}^{-1}\text{]}$$

$$\lambda_2 = 0,09 \text{ [W.m}^{-1}\text{.K}^{-1}\text{]}$$

Po doplnění do vzorce vyšlo:

$$K = 3,458 \text{ [W.m}^{-2}\text{.K}^{-1}\text{]}$$

Odtud tedy mohou dosadit do bilanční rovnice:

$$P = 0,02095 \cdot 1,29 \cdot 1000 \cdot \frac{dt}{dT} + 0,01 \cdot 628 \cdot 1250 \cdot \frac{dt}{dT} + 3,458 \cdot 0,458 \cdot (t_0 - t_1)$$

Pomocí výše uvedené bilanční rovnice mohou nyní provádět výpočty teploty a času. Tyto by měly stačit pro ověření fyzikálních vlastností tepelné soustavy.

10.1.3 Maximální dosažitelný rozdíl teplot dané tepelné soustavy

Tento rozdíl se vypočte pomocí bilanční rovnice. Za topný výkon se zvolí 60W. Vyjádří se Δt . Takže postup vypadá takto:

$$60 = \Delta t \cdot \left(\frac{27,0255}{\Delta T} + \frac{7850}{\Delta T} + 1,583 \right)$$

$$\text{pro } \Delta T \rightarrow \infty \text{ platí: } \lim_{\Delta T \rightarrow \infty} \left(\frac{27,0255}{\Delta T} + \frac{7850}{\Delta T} + 1,583 \right) = 1,583$$

$$60 = \Delta t \cdot 1,583$$

$$\Delta t = \frac{60}{1,583}$$

$$\Delta t = 37,902^\circ\text{C}$$

Z výše uvedeného výpočtu plyne, že maximální teplotní rozdíl, který může být dosažen činí 37,902 °C.

10.1.4 Čas potřebný pro zvýšení teploty soustavy o jeden stupeň

Tento čas se stanoví při teplotě okolního vzduchu 20 °C. Upraví se bilanční rovnice, dosadí za $\Delta t = 1$ a vytkne se ΔT . Samotný výpočet pak vypadá takto:

$$60 = 27,0255 \cdot \frac{1}{\Delta T} + 7850 \cdot \frac{1}{\Delta T} + 1,583 \cdot 1$$

$$\Delta T = \frac{27,0255 + 7850}{60 - 1,583}$$

$$\Delta T = 134,84 \text{ s}$$

Z výše uvedeného postupu vyplývá, že systém se ohřeje o jeden stupeň celsia za 134,84 sekund. Pomocí tohoto postupu lze pak lehce spočítat, za jakou dobu se systém ohřeje o

libovolný teplotní rozdíl. Při překročení maximálního rozdílu teplot, tj. 37,902 °C, nedává tento vzorec reálný čas (vrací záporné hodnoty).

10.1.5 Čas potřebný pro zvýšení teploty o maximální rozdíl

Tento čas bohužel nejde určit, protože při výpočtu času bude porušena podmínka nenulového jmenovatele a dojde k dělení nulou (čas je pro toto řešení tedy nekonečný). Přesto je možné určit rozumný čas, za který se povede systému dosáhnout maximálního teplotního rozdílu zmenšeného o 1% - tedy $37,902 - 1\% = 37,5228$.

Pro tuto teplotu tedy potom vyjde čas $t = 491460$ sekund.

10.2 Testování regulátoru

Při testování regulátoru, byl kontrolován jím dosažený čas, při stanovené teplotní odchylce. Testovací odchylka byla stanovena na 7 °C, tedy regulátor dostal za úkol zvýšit teplotu v soustavě o tuto hodnotu za co nejlepší časový úsek. Pro kontrolu jsem použil vzorec z kapitoly 10.1.4, „Čas potřebný pro zvýšení teploty o jeden stupeň“.

Nejprve se provede výpočet a poté se provede kontrola reálné soustavy proti tomuto výsledku.

10.2.1 Výpočet nejrychlejšího času pro dosažení zvýšení teploty soustavy o 7 °C

$$60 = 27,0255 \cdot \frac{7}{\Delta T} + 7850 \cdot \frac{7}{\Delta T} + 1,583 \cdot 7$$

$$\Delta T = \frac{7 \cdot (27,0255) + 7 \cdot (7850)}{60 - (7 \cdot 1,583)}$$

$$\Delta T = 1127,15 \text{ s} = 18 \text{ min } 47 \text{ sec } 15 \text{ sec} / 100.$$

Tedy z výsledku vyplývá, že nejideálnější čas, který by šel pro zvýšení teploty o 7 °C dosáhnout, je asi 18 minut a 47 sekund. Při dosažení této teploty přijme tepelná soustava od topného tělesa celkové teplo ve výši 67,62 kJ.

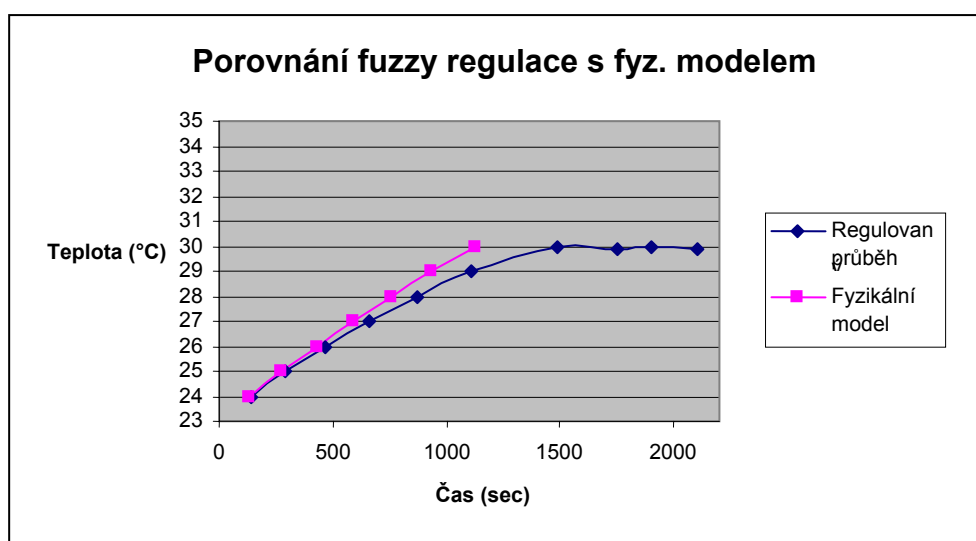
10.2.2 Testování regulátoru

Regulátor dostal za úkol zvýšit teplotu o 7°C. Při tomto pokusu jsem ověřil, že teplota okolí byla při zahájení testu a i po celou dobu 23 °C. Žádaná teplota byla tedy nastavena na 30 °C. Regulátor pracoval podle stanoveného algoritmu. Výsledky jsou v tabulce č. 6.

Tab. č. 6 – Naměřené reálné hodnoty

Teplota (°C)	Čas (s)
23	0
24	137
25	291
26	464
27	656
28	870
29	1109
30	1490

Z tohoto pokusu vyplývá, že regulátor dosáhl stanovené teploty za 1490 sekund, tedy za 24 minut a 50 vteřin – viz. tabulka č. 6. Byl tedy o 363 sekund pomalejší, než fyzikální model. Musím však podotknout, že při regulaci nedošlo k překmitu a po dosažení žádané teploty již regulátor stanovenou teplotu udržel.



Obr. č. 38 – Graf porovnání fuzzy regulace s fyzikálním modelem

Dále z testu jednoznačně vyplývá, že při bezpřekmitové regulaci bylo protopeno navíc $363 \cdot 60 \text{ J} =$ tedy 21,78 kJ, což je asi o 32,2 % víc potřebného tepla oproti fyzikálnímu modelu. Grafické porovnání fuzzy regulace s fyzikálním modelem viz. obr. č. 38.

10.2.3 Testování ventilace, úloha ventilace v systému

Při tomto testu šlo o to, zjistit, o jakou teplotu se zvedne teplota v systému, jestliže se zároveň s topným tělesem spustí také ventilátor. Topení probíhalo po celou dobu na 100%, nebylo tedy nijak regulováno.

Test měl probíhat 10 minut. Bylo potřeba zjistit, jaký vliv má na regulovanou soustavu ventilátor. Při testu byla aktuální teplota okolí 25,6 °C a během něj byla pořízena níže uvedená data:

Tab. č. 7 – Hodnoty naměřené při testu ventilace

Teplota (°C)	Čas (s)
25,6	60
26,1	120
26,6	180
27,2	240
27,5	300
27,9	360
28,3	420
28,6	480
28,8	540
29,0	600

Za deset minut se teplota v soustavě zvýšila o pouhé 4,6 °C. Již z naměřených hodnot je vidět, že teplota stoupala pomaleji než při regulaci nebo při ideálním fyzikálním modelu, proto se domnívám, že ventilátor působí na teplotu negativně, tedy mísí rychleji venkovní vzduch se vzduchem uvnitř.

10.2.4 Zhodnocení výsledků testu

Podle mého názoru, fuzzy regulace proběhla úspěšně, neboť i při horším dosaženém času regulace oproti fyzikálnímu modelu regulátor pracoval bez překmitu. Nebylo v mých si-

lách možné zajistit ekonomickou stránku pokusu současně s požadavkem na precizní regulaci. S největší pravděpodobností byla časová prodleva způsobena opatrnějším dodáváním tepla do soustavy a určitě také tepelnými ztrátami.

Dosažené výsledky mne uspokojily, avšak věřím tomu, že celkovou optimalizací systému by mohlo dojít ještě k lepším výsledkům. Určitě by pomohlo odstranění ventilátoru a ventilačních průduchů. Pomohlo by také instalovat tepelnou izolaci a lépe utěsnit víko krabice.

ZÁVĚR

Cílem této diplomové práce bylo vypracovat literární rešerši zaměřenou na problematiku vzdáleného monitorování a řízení technologického procesu pomocí mobilní sítě. Vzhledem k tomu, že pro úspěšné splnění úkolu stačilo pouze vytvořit simulaci regulace a ovládat tuto simulaci pomocí GSM zařízení, myslím si, že moje řešení zadání nejen že splnilo, ale také určitě překonalo.

Teoretickou část této práce jsem věnoval historii, vzniku a vývoji GSM sítí. Dále jsem se věnoval problematice embedded zařízení, neboť jsem jedno zařízení pro vypracování mé diplomové práce využil. V poslední části jsem se věnoval fuzzy logice, která byla obsažena v software regulátoru.

Fuzzy logiku jsem si vybral proto, neboť navržený technologický proces měl být řízen co nejjednodušeji a jeho výroba neměla trvat příliš dlouho. A právě fuzzy logika mi umožnila splnit právě tyto výše uvedené předpoklady.

Při sestavování praktické části jsem dbal na logický sled uváděných informací, a snažil jsem se o co nejpřesnější popis návrhu této monitorovací a řídicí soustavy. Dále jsem věnoval značnou část této sekce také popisu vzniku a výroby reálné monitorovací a řídicí soustavy. Jelikož současně s tímto dokumentem vznikala i reálná část, bude tato součástí prezentace a obhajoby diplomové práce.

Diplomovou práci jsem zakončil fyzikálním popisem reálné tepelné soustavy a jednoduchým otestováním navržené fuzzy regulace. Dosažený výsledek dokazuje jak teoretickou, tak i praktickou upotřebitelnost všech výše uvedených technologií k návrhu a realizaci libovolných monitorovaných a řízených technologických procesů.

Doufám, že tato práce splnila svůj účel – tedy že jsem pomocí ní byl schopen ukázat praktické využití mobilních sítí a embedded zařízení. Věřím tomu, že tato diplomová práce bude inspirovat řadu čtenářů v jejich vlastním výzkumu a tvorbě.

SEZNAM POUŽITÉ LITERATURY

- [1] BALÁTĚ, J. Automatické řízení. 1. vyd. Praha: BEN, 2003. 664 s. ISBN 80-7300-020-2.
- [2] ĎADO, S. KREIDL, M. Senzory a měřicí obvody. Praha: ČVUT, 1999.
- [3] CHUDÝ, V., PALNEČÁR, R. Meriania technických veličin. Bratislava: STU Bratislava, 1999.
- [4] PUŽMAN, J. Dálkový přenos dat. Praha: Vydavatelství Grada, 1985.
- [5] ZEZULKA, F., FIEDLER, P. Ethernet v průmyslové automatizaci. In: Sborník referátů z doprovodné konference Pragoregula 2000, Praha, 2000.
- [6] ADVANTECH Co. Your ePlatform Partner [online]. c1983-2004 [cit. 2004-06-03]. Dostupný z: <<http://www.advantech.com>>
- [7] BOLDIŠ, P. Citace dokumentů podle norem ISO 690 a ISO 690-2e [online]. Poslední revize 1.12.2003 [cit. 2004-06-03]. Dostupný z: <<http://www.boldis.cz/citace/citace.html>>
- [8] MOTOROLA Inc. Glosář [online]. Dostupný z:<http://direct.motorola.com/cze/glossary.asp?index_key=G&country=CZE&language=CZE>
- [9] STRNAD, P. Historie GSM [online]. Poslední revize 1.12.1999 [cit. 2004-06-03]. Dostupné z: <<http://www.volny.cz/drd/gsm/GSMhistory.html>>
- [10] WWW.MOBIL.IDNES.CZ. Stručný úvod do GSM [online]. c2004 [cit. 2004-06-03]. Dostupný z: <http://mobil.idnes.cz/mobilni_komunikace/mobilni_technologie/GSM/c1-1.html>
- [11] MobilMania.cz – Co je to EDGE? Stručně a jasně [online]. Poslední revize: 27.11.2003. Dostupný z: <www.mobilmania.cz/Profi/AR.asp?ARI=105934>
- [12] Kéhar, O. Iridium [on-line]. Poslední revize 22.7.2002 [cit. 2004-06-03]. Dostupný z: <<http://web.quick.cz/astrokoutek/iridium/iridium.htm>>
- [13] Snášel, J. GPRS pod mikroskopem (I. Část) [on-line]. Poslední revize 12.5.2004 [cit. 2004-06-03]. Dostupný z: <<http://www.mobilmania.cz/Profi/AR.asp?ARI=107243&CAI=2119>>

- [14] Snášel, J. GPRS pod mikroskopem (II. část) [on-line]. Poslední revize 14.5.2004 [cit. 2004-06-03]. Dostupný z: <<http://www.mobilmania.cz/Profi/AR.asp?ARI=107257>>
- [15] NOVÁK, V. Základy fuzzy modelování. 1. vyd. Praha: BEN, 2000. ISBN 80-7300-009-1
- [16] Matsushita Electric Works, Ltd. Home Page [on-line]. ©2004 [cit. 2004-06-03]. Dostupný z: <<http://www.mew.co.jp/e-index.html>>
- [17] Wikipedia. x86 [online]. Poslední revize 3.5.2004 [cit. 2004-06-03]. Dostupný z: <<http://en.wikipedia.org/wiki/X86>>
- [18] Wikipedia. Acorn RISC Machine [online]. Poslední revize 6.5.2004 [cit. 2004-06-03]. Dostupný z: <http://en.wikipedia.org/wiki/Advanced_RISC_Machine>
- [19] Wikipedia. PowerPC [online]. Poslední revize 11.5.2004 [cit. 2004-06-03]. Dostupný z: <<http://en.wikipedia.org/wiki/PowerPC>>
- [20] Wikipedia. MIPS architecture [online]. Poslední revize 6.6.2004 [cit. 2004-06-07]. Dostupný z: <http://en.wikipedia.org/wiki/MIPS_architecture>
- [21] BDTI. Hitachi SH3-DSP [online]. [cit. 2004-06-07]. Dostupný z: <<http://www.bdti.com/procsum/sh3-dsp.htm>>
- [22] Disclaimer. Scalable Procesor ARChitecture [online]. c2004 [cit. 2004-06-07]. Dostupný z: <<http://computing-dictionary.thefreedictionary.com/Scalable%20Processor%20ARChitecture>>
- [23] WIKIPEDIA. RISC [online]. Poslední revize 6.6.2004 [cit. 2004-06-07]. Dostupný z: <http://en.wikipedia.org/wiki/Reduced_Instruction_Set_Computer>
- [24] WIKIPEDIA. Unix [online]. Poslední revize 4.6.2004 [cit. 2004-06-07]. Dostupný z: <<http://en.wikipedia.org/wiki/UNIX>>
- [25] WIKIPEDIA. Linux [online]. Poslední revize 9.6.2004 [cit. 2004-06-10]. Dostupný z: <<http://en.wikipedia.org/wiki/Linux>>
- [26] WIKIPEDIA. Symbian OS [online]. Poslední revize 30.5.2004 [cit. 2004-06-10]. Dostupný z: <http://en.wikipedia.org/wiki/Symbian_OS>

- [27] WIKIPEDIA. Embedded system [online]. Poslední revize 2.6.2004 [cit. 2004-06-10]. Dostupné z: <http://en.wikipedia.org/wiki/Embedded_system>
- [28] MICROSOFT CORPORATION. MSDN Home Page [on-line]. c2004 [cit. 2004-06-10]. Dostupný z: <<http://msdn.microsoft.com>>
- [29] MERUNKA, M. Satelitní síť Iridium může výrazně zvýšit bezpečnost letecké dopravy [on-line]. Poslední revize 10.10.2001 [cit. 2004-06-07]. Dostupný z: <<http://www.isdn.cz/clanek.php?cid=3259>>
- [30] Microsoft® Visio 2003 Product Information [on-line]. Poslední revize 7.6.2004 [cit. 2004-06-07]. Dostupný z: <<http://www.microsoft.com/office/visio/prodinfo/default.mspx>>
- [31] PAPOUCH, s.r.o. Digitální teplotní čidlo se sériovou komunikací [on-line]. Poslední revize 7.6.2004 [cit. 2004-06-07]. Dostupný z: <<http://www.papouch.com/tm.html>>
- [32] UMTS Forum [online]. Poslední revize 29.8.2006 (denně). Dostupný z: <<http://www.umts-forum.org>>
- [33] iDNES.cz - UMTS už klepe na dveře. Co všechno umí? Poslední revize 7.10.2005 [cit. 2006-08-29]. Dostupný z: <http://mobil.idnes.cz/mob_tech.asp?r=mob_tech&c=A051006_220629_mob_tech_dno>
- [34] Advantech – Trusted ePlatform Services – PCM-5822 [online]. Poslední revize 28.8.2006. Dostupný z: <http://www.advantech.com/products/Model_Detail.asp?model_id=1-1TGZM2>
- [35] M75 – BenQ-Siemens Mobile Phones Portal [online]. Poslední revize 28.8.2006. Dostupný z: <http://www.benq-siemens.com/cds/frontdoor/0,2241,hq_en_0_130806_0,00.html>
- [36] Rapid application development – Wikipedia, the free encyclopedia [online], Poslední revize: 28.8.2006. Dostupný z: <http://en.wikipedia.org/wiki/Rapid_application_development>
- [37] Výroba plošných spojů fotocestou [online]. Poslední revize: 10.2.2002. Dostupný z: <<http://www.volny.cz/mavit2/Elektro/Fotocesta/fotocesta.html>>

- [38] Windows History: Windows Desktop Products History [online]. Poslední revize: 30.6.2002. Dostupný z: <<http://www.microsoft.com/windows/WinHistoryDesktop.mspx>>
- [39] PHP: Hypertext Preprocessor [online]. Poslední revize: 29.8.2006. Dostupný z: <<http://www.php.net>>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

Zkratky jednotek

bps	bit per second – bit za sekundu
kbps	kilobit per second – 1024 bitů za sekundu
Mbps	Megabit per second - 1 048 576 bitů za sekundu
B	Byte = 8 bitů
kB	kilo Byte = 1024 Byte = 8192 bitů
MB	Mega Byte = 1024 kilo Byte = 8 388 608 bitů
GB	Giga Byte = 1024 Mega Byte = 1 048 576 kilo Byte = $8,589 \times 10^9$ bitů
TB	Tera Byte = 1024 Giga Byte = 1 048 576 Mega Byte = $8,796 \times 10^{12}$ bitů

Zkratky názvů

AMD	American Micro Devices
AMPS	Advanced Mobile Phone Service
BIOS	Basic Input Output System
BOS	Basic Operation System
BSC	Base Station Controller
BTS	Base Transceiver Station
CEPT	Conference of European Post and Telecommunication
CS	Code Scheme
CSD	Circuit Switched Data
DCS	Data Coding Scheme
DoV	Data over Voice
DOS	Disk Operation System
ESCD	Extended System Configuration Data

EDGE	Enhanced Data for GSM Evolution
FCC	Federal Communications Commission
FDD	Frequency Division Duplex
GGSN	Gateway GPRS Support Node
GTP	GPRS Tunneling Protocol
GSM	Global System for Mobile communication
GPRS	General Packet Radio Service
GUI	Graphical User Interface
HTTP	Hyper Text Transfer Protocol
HCF	Halt and Catch Fire
HSCSD	High-Speed Circuit-Switched Data
IOCTL	Input Output Control
LCD	Liquid Crystal Display
LSI	Large Scale Integration
MSI	Medium Scale Integration
MoU	Memorandum of Understanding
MTBF	Mean Time Between Failure
PCS	Personal Communications Services
PHP	PHP Hypertext Preprocessor
RMA	Rate Monotonic Analysis
RNS	Radio Network System
RNC	Radio Network Controller
SSI	Small Scale Integration
TDD	Time Division Duplex
TOS	Tape Operation System

UMTS Universal Mobile Telecommunication System

UNICODE Universal Coding scheme

WAP Wireless Application Protocol

WCDMA Wideband Code Division Multiple Access

WWW World Wide Web

SEZNAM OBRÁZKŮ

Obr. č. 1 – Základní schéma GPRS	19
Obr. č. 2 – Porovnání modulací GPRS vs. EDGE	21
Obr. č. 3 – TFT Panel PC Kit - PPC 1211T	26
Obr. č. 4 – Nosná raketa projektu Apollo 11	27
Obr. č. 5 – Lunární modul Apolla 11	28
Obr. č. 6 – Odkrytý korpus nukleární střely Minuteman I s pohledem na řídicí elektroniku	28
Obr. č. 7 – Intel 4004	29
Obr. č. 8 – Intel 8080A	30
Obr. č. 9 – Motorola XC6800A	31
Obr. č. 10 – Původní Intel 8051 ve 40 pinovém pouzdře	31
Obr. č. 11 – JS2051 – Intel 8051 kompatibilní	32
Obr. č. 12 – AMD 5x86	33
Obr. č. 13 – Advantech PC/104 Plus CPU module PCM 3370	34
Obr. č. 14 – Logo Embedded Linux Consortium	38
Obr. č. 15 – Logo Microsoft ® Windows Embedded	39
Obr. č. 16 – Logo Symbian OS	39
Obr. č. 17 – Modulární schéma řešení	58
Obr. č. 18 – Vzhled desky Biscuit PC Advantech ® PCM-5822	61
Obr. č. 19 – Rozložení součástí na základní desce Advantech ® PCM-5822	63
Obr. č. 20 – Teplotní čidlo TM od firmy PaPouch® Elektronika	67
Obr. č. 21 – Siemens M75	68
Obr. č. 22 – Datový kabel DCA 500	68
Obr. č. 23 – časová linie vývoje operačních systémů od společnosti Microsoft	69
Obr. č. 24 – Pohled na krabici shora	82
Obr. č. 25 - Vlevo pohled na ventilátor se spínačem, vpravo pohled na odkrytované ventilační díry	82
Obr. č. 26 - Pohled na otevřenou krabici shora	83
Obr. č. 27 – Pohled na svorkovnici (vlevo) a na síťový vypínač (vpravo)	84
Obr. č. 28 – Zdroj stejnosměrného napětí	85
Obr. č. 29 – Žárovka v objímce	85

Obr. č. 30 – Pohled na ventilátor zevnitř	86
Obr. č. 31 – Pohled na spínač ventilátoru	86
Obr. č. 32 – Pohled na přepínač režimů ventilátoru	86
Obr. č. 33 – Pohled na Relay Desk v1.0	86
Obr. č. 34 – Pohled na Com Switch v 1.0.....	87
Obr. č. 35 – Pohled z boku a shora na uchycené Bisucit PC Advantech ® PCM-5822.....	87
Obr. č. 36 – Grafické znázornění reálné soustavy	93
Obr. č. 37 – Zobrazení řezu dřevotřískou	95
Obr. č. 38 – Graf porovnání fuzzy regulace s fyzikálním modelem.....	101

SEZNAM TABULEK

Tab. č. 1 – Třídy GPRS zařízení.....	18
Tab. č. 2 – Přenosové rychlosti služeb GSM a GPRS	19
Tab. č. 3 – Porovnání GPRS vs. EDGE.....	21
Tab. č. 4 – Složení atmosférického vzduchu	56
Tab. č. 5 – Adresování zařízení pomocí Com Switch v1.0	66
Tab. č. 6 – Naměřené reálné hodnoty	101
Tab. č. 7 – Hodnoty naměřené při testu ventilace	102

SEZNAM ROVNIC

Rovnice č. 1 – P-regulátor klasický	50
Rovnice č. 2 – P-regulátor fuzzy	50
Rovnice č. 3 – PD-regulátor klasický	50
Rovnice č. 4 – PD-regulátor fuzzy.....	50
Rovnice č. 5 – PI-regulátor klasický.....	50
Rovnice č. 6 – PI-regulátor fuzzy	51
Rovnice č. 7 – PID-regulátor klasický.....	51
Rovnice č. 8 – PID-regulátor fuzzy	51

ABECEDNÍ REJSTŘÍK

3	
3GPP.....	15
A	
AMD.....	33
AMPS.....	12
ANSI C/C++.....	26
Apache WWW.....	71
API.....	42, 43
architektura	
RISC.....	36, 37
B	
BIOS.....	26
BIOS - AWARD.....	62
Biscuit PC.....	58, 60, 61
Advantech PCM-3370.....	34
Advantech PCM-5822.....	66
BOS.....	27
breakpoint.....	40
C	
CarryOver minutes.....	15
CD-ROM.....	35
CD-RW.....	35
CEPT.....	13
CMOS.....	35
Collaborate Object Model.....	71
CPU.....	45
AMD 5x86.....	33
AMD Athlon 64.....	35
AMD Geode™ processor.....	62
Cyril.....	33
Intel 80386.....	32
Intel 80486.....	33
Intel 8051.....	31
Intel 8080.....	30
Intel 8086.....	35
Intel Pentium IV.....	35
mikrokód.....	40
D	
DCS.....	14
defuzzifikace.....	52
DIMM PC.....	4, 89
DOS.....	27
DVD-ROM.....	35
E	
EDGE.....	20, 22
8PSK.....	20
ECSA.....	21
HSCSD.....	21
MCS1.....	20
MCS9.....	20
Push-to-talk.....	22
embedded.....	25, 27, 41
AMD.....	47
Apollo Guidance Computer.....	27, 28
flash paměť.....	25
GUI.....	25
HCF.....	30
Hitachi.....	47
Charles Stark Drapper.....	27
IBM.....	47
image.....	41
Intel.....	47
Intel 8051.....	31
key-pad.....	25
MIT.....	27
Motorola.....	47
Motorola 6800.....	30
PC104.....	26
PDP-11.....	40
SWTPC 6800.....	30
Vectrex.....	30

VIA	47	GTP.....	20
Windows CE 2.11	26	HLR.....	20
Zilog.....	31	IWU	19
embedded debugging.....	40	PCU	20
embedded operační systémy		QoS	18
Control Loop OS.....	42	SGSN.....	20
Exotic OS.....	45	GSM	10, 13, 60
Nonpreemptive multitasking OS.....	43	bandwidth	15
Office-style OS	44	BSC.....	12
Preemptive tasks OS	44	BTS.....	12
Preemptive timers OS	44	MMS.....	16
Embedded PC	4, 35, 61, 89	PSTN	19
Ethernet 10-Base2.....	59	SMSC.....	19
ETSI.....	14, 16		
F		H	
fáze kontroly návrhu	92	HSCSD	15
fáze realizace	93	HTTP metody	
fáze testování	93	GET	79
fáze vytváření návrhu	92	POST	79
FCC.....	14	HTTP request	79
firmware.....	26	I	
fuzzy logika	48, 49	IBM	27, 34, 36
E. H. Madani.....	48	IC čipy	26
L.A. Zadeh	48	image	41, 42
fuzzy model	53	Interrupt.....	42
fuzzy regulace.....	49, 50	IOCTL	45
fuzzy regulátor.....	49, 50, 74	IPC.....	26
parametry	52	Iridium.....	24
PD-regulátor.....	51	Iridium Satelite LLC.....	24
PID-regulátor	52	K	
PI-regulátor	51	kernel core	43
P-regulátor	51	key-pady	25
G		L	
GPRS	16	LA.....	12
CS.....	17	LCD monitor	25
CSD.....	16	M	
DoV.....	16	Microsoft Windows CE .NET	39
GGSN.....	20		

mikrokontrolér	35, 40	Andrew Tanenbaum.....	37
Motorola	40	architektura	37
Motorola 6800	30	op-kód instrukce	37
MTBF	26	RMA	45
N		ROM.....	26, 28, 35, 41
NASA	27, 29	RS-232.....	59
Nintendo	36	S	
Nokia	40	S. fuzzy-logika	
O		Assilian	48
operační systém		Samsung	40
Linux	38	sběrnice	
Microsoft Windows CE	39	EIDE	62
Symbian OS	39	EISA	34
UNIX	38	ISA.....	34, 35
P		PC/104	34, 35
Panasonic	40	PC/104 Plus	35
PC/104	62	PCI	32, 34
PCS	14	PCI-X.....	32
PHP.....	79	SCSI.....	33
platforma		USB	34
ARM, Strong ARM.....	36	USB 2.0	35
HPC.....	36	sériový kabel DCA-500.....	69
Intel x86	35	server	
MIPS	36	GSM server.....	59
PPC	36	TCP/IP telnet server.....	59
SH1 až SH4.....	37	WWW server	59
SPARC.....	37	sestavení TCP/IP spojení	
x86	35	accept.....	79
Plug & Play.....	34	bind.....	79
Power Management	62	listen.....	79
pracovní cyklus.....	91	schéma soustavy	58
Přepínač sériových portů	89	Siemens	40
R		Siemens M75	68
RAM	28, 32, 33, 35, 41, 42	signály sériového portu.....	67
regulátor.....	4	Sinclair ZX Spektrum.....	31
Reléová deska	88	Socket 478	64
RISC		Sony-Ericsson.....	40
		STAND-BY.....	28
		system core	43

SYSTEM/360	27	LSI	29
system-on-chip.....	36	MSI	29
T		SSI	26
TCP-IP	4	V	
tepelně bilanční rovnice	98	ventilátor.....	87
teplotní čidlo TM	67	Ventilátor	87, 126
timer	42, 44	VoiceMail.....	15
timeslot	13	výroba plošného spoje	81
topné těleso	58	cuprexitit	81
TOS.....	27	hydroxid sodný	81
U		kyselina chlorovodíková.....	81
UART	31	oxid železitý.....	81
UMTS	22	peroxid vodíku.....	81
BS.....	23	UV záření.....	81
BSC.....	23	vzduch	57, 104
FDD	22, 23	W	
handover.....	23	WAP	15
RNC	23	Windows CE 3.0.....	41
RNS.....	23	WWW	4
TDD	22	Z	
UMTS R4.....	24	zdroj stejnosměrného proudu	87
UMTS R99.....	24	znaková sada	
UMTS-UTRAN.....	23	ANSI.....	72
videotelefonie.....	24	UNICODE	72
WCDMA.....	22		
úrovně integrace			

SEZNAM PŘÍLOH

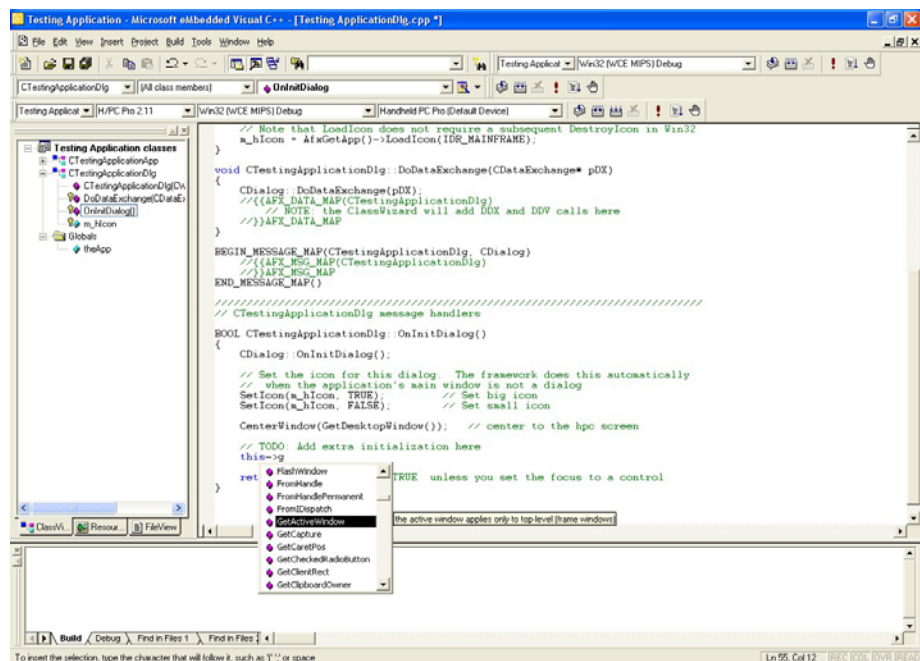
- P I Přehled služeb poskytovaných v rámci GPRS
- P II Ukázka vývojového prostředí Microsoft Windows CE Tools 3.0
- P IIIA PC/104 single-board PC, Advantech PC/104-Plus CPU module PCM 3370
- P IIIB PC/104 single-board PC, Advantech PC/104-Plus CPU module PCM 3370
- P IVA Schéma PC/104 sběrnice a význam jejích pinů
- P IVB Schéma PC/104 sběrnice a význam jejích pinů
- P V Základní informace o dřevotříse
- P VI Seznam příkazů použitelných pro GSM Server
- P VII Seznam příkazů použitelných pro Telnet Server
- P VIII Blokové schéma fuzzy regulátoru
- P IX Blokové schéma GSM Serveru
- P X Blokové schéma Telnet Serveru
- P XI Vzhled návrhu desky Relay Desk v1.0, schéma zapojení.
- P XIIA Vzhled návrhu desky Com Switch v1.0
- P XIIB Vzhled návrhu desky Com Switch v1.0
- P XIII Vzhled návrhu desky Com Switch v1.0

PŘÍLOHA P I: PŘEHLED SLUŽEB POSKYTOVANÝCH V RÁMCI GPRS

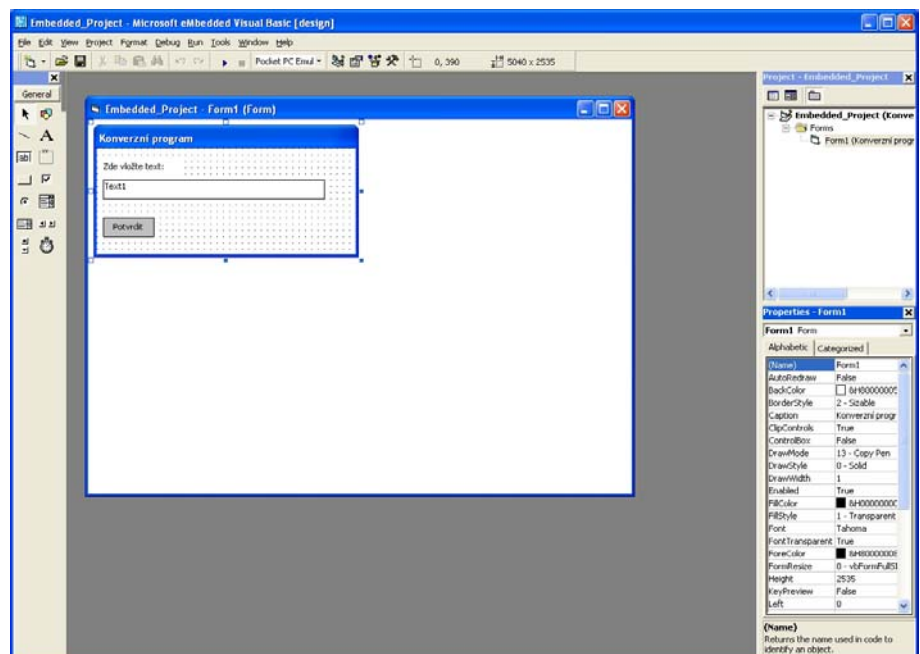
Kategorie služeb	Služba	Komentář
Telematické služby	Half-rate kódování řeči (HR)	volitelná implementace dovolující použití dual rate (half rate a full rate) v jednom telefonu.
	Enhanced full-rate kódování řeči (EFR)	volitelná implementace dovolující použití dual rate (enhanced full rate a full rate) v jednom telefonu.
Doplňkové služby	Calling line identification (CLI)	zobrazení a zamezení zobrazení čísla volajícího
	Connected line identification (COL)	Zobrazení a zamezení zobrazení čísla volaného.
	Call waiting (CW)	v průběhu aktivního hovoru informuje o jiném příchozím hovoru - používá se s funkcí call hold.
	Call hold (CH)	"zaparkuje" probíhající hovor a umožní nový hovor/příjem jiného hovoru. Používá se s funkcí call waiting.
	Multiparty communication (MPTY)	až pět hovorů může být spojeno do jednoho telekonferenčního hovoru.
	Closed user group (CUG)	obdoba funkce z trunku.
	Advice of Charge (AoC)	informace o ceně hovoru.
	Unstructured supplementary services data (USSD)	umožní operátorovi tvorbu individuálních služeb.
	Operator-determined barring (ODB)	operátorem definovaná omezení skupiny uživatelů.

PŘÍLOHA P II: UKÁZKA VÝVOJOVÉHO PROSTŘEDÍ MICROSOFT WINDOWS CE TOOLS 3.0

a) Vzhled programu Microsoft Embedded Visual C++ 3.0

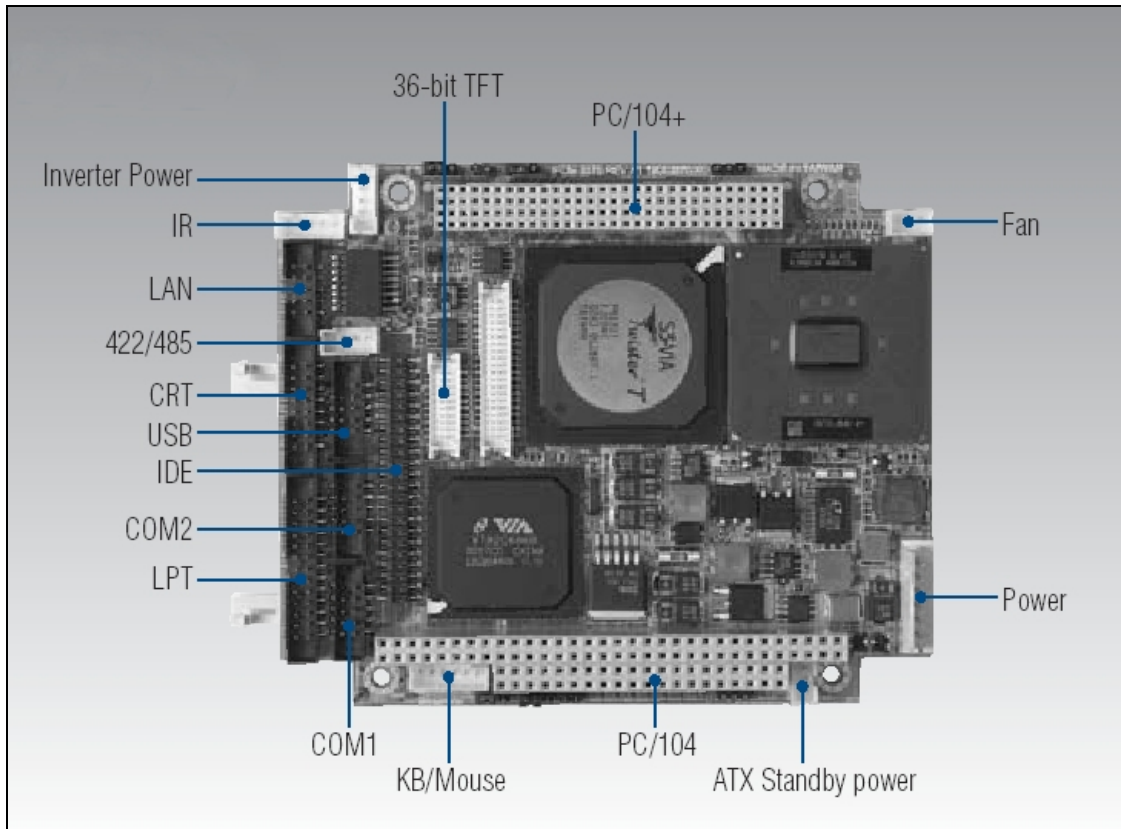


b) Vzhled programu Microsoft Embedded Visual Basic 3.0



PŘÍLOHA P IIIA: PC/104 SINGLE-BOARD PC, ADVANTECH PC/104-PLUS CPU MODULE PCM 3370

Vzhled a rozložení součástí na PC/104 desky (anglicky)



PŘÍLOHA P IIIB: PC/104 SINGLE-BOARD PC, ADVANTECH PC/104-PLUS CPU MODULE PCM 3370

Specifikace desky:

- CPU Onboard ULV Celeron® 400/650 bez ventilátoru, LV Pentium® III 933(800 MHz doporučená frekvence)
- Systémový chipset VIA VT8606/TwiserT + VT82C686B
- BIOS Award 256KB Flash BIOS
- Second level cache 256Kb na procesorech Intel®
- DRAM 1 x SODIMM socket, podporuje až 512MB SDRAM
- SSD Supports CompactFlash Card Type I
- Podporuje řízení spotřeby APM (Advanced Power Management)
- PCI IDE rozhraní 1 x Enhanced IDE interface
 - 1st rozhraní podporuje 2 IDE zařízení (1 Master, 2 Slave)
 - PIO režim 3, 4 vybavený Bus Masteringem až do rychlosti 14 MB/s
- 1 x paralelní port, podporuje režimy EPP/ECP
- 2x sériový port + 5V jumper
 - COM1: RS-232
 - COM2: RS-232/422/485 (422/485 kabely jsou volitelné)
 - IrDA podpora, přenosová rychlost až 1.15 Mbps
- Watchdog timer hardware 1.6 sec, jde však změnit pomocí software, tento generuje systémový reset nebo IRQ 11
- Klávesnice/myš - 6-pin Mini-Din konektor konektor pro klávesnici a PS/2 myš
- USB 2 (Universal Serial Bus) – zpětně kompatibilní s verzí 1.1
- Ventilátor konektor (VCC +5V), kontrola rychlosti otáček
- Systémový monitor pro teplotu čipové sady a mikroprocesoru
- PC/104, PC/104+ a PCI PC/104-Plus module connector

- ethernet adaptér

Chipset Realtek RTL8139D 10/100Mbps, Ethernet interface RJ-45, 10/100Base-T

- řadič grafické karty

Nachází se na čipsetu VIA VT8606 obsahuje 4 x AGP kontrolér, podporuje tyto typy zobrazovacích jednotek analog CRT; 36, 24, 18 bit TFT; 24-bit DSTN (volitelné)

Balení obsahuje:

- 1 PCM-3370 deska,

- kabel ke klávesnici/myši Y-Cable (p/n:1700060202),

- kable k externí Y-Cable (p/n:1703060053),

- kabel k VGA Cable (p/n:1701160150),

- kabel k Ethernet RJ-45 konektoru (p/n:1701100202),

- kabel k IDE (p/n:1701440350),

- kabel k sériovému portu (p/n:1700100250),

- kabel k paralelnímu portu (p/n:1700260250),

- napájecí ATX kabel (p/n:1703200380) ,

- napájecí AT kabel (p/n:1703080104),

- tištěný manuál,

- CD-ROM (Manuály, Ovladače, Utility),

- kabel k 422/485 (p/n:1703040257).

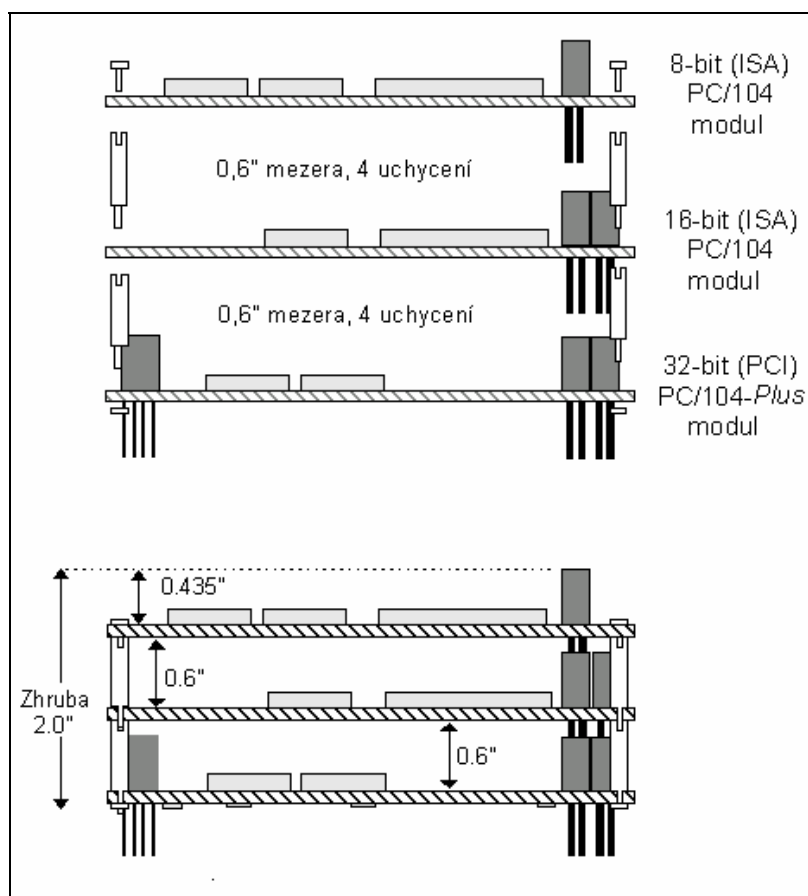
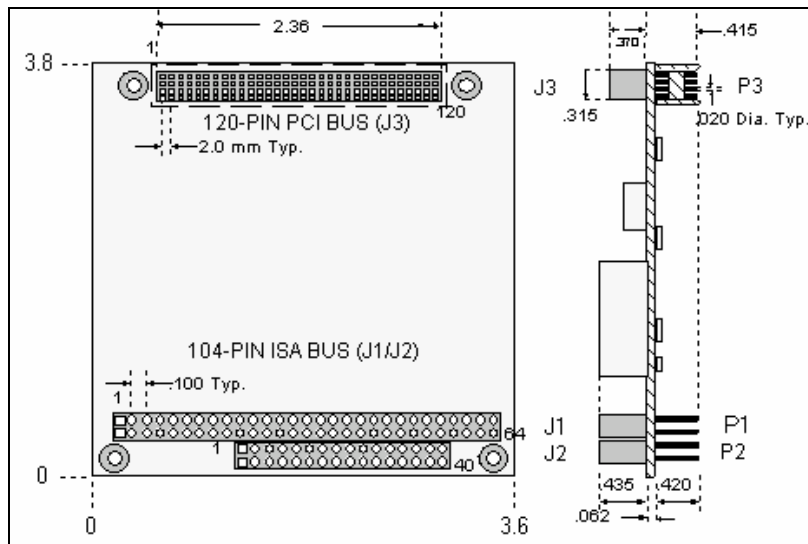
Rozměry a pracovní teplota

- Velikost desky 96 mm x 115 mm,

- Pracovní teplota 0 ~ 60 °C.

PŘÍLOHA P IVA: SCHÉMA PC/104 SBĚRNICE A VÝZNAM JEJÍCH PINŮ

Vzhled desky PC/104-Plus + ukázka montáže



PŘÍLOHA P IVB: SCHÉMA PC/104 SBĚRNICE A VÝZNAM JEJÍCH PINŮ

Význam jednotlivých pinů na sběrnici

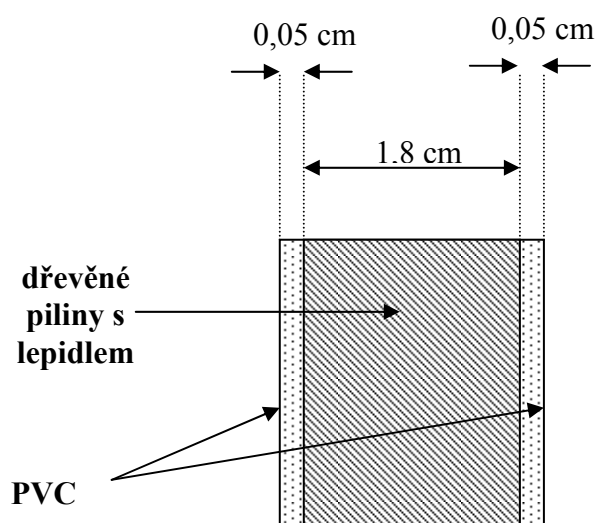
J3/P3				
Pin	A	B	C	D
1	GND	Reserved	+5	AD00
2	V/I/O	AD02	AD01	+5V
3	AD05	GND	AD04	AD03
4	C/BE0#	AD07	GND	AD06
5	GND	AD09	AD08	GND
6	AD11	V/I/O	AD10	M66EN
7	AD14	AD13	GND	AD12
8	+3.3V	C/BE1#	AD15	+3.3V
9	SERR#	GND	Reserved	PAR
10	GND	PERR#	+3.3V	Reserved
11	STOP#	+3.3V	LOCK#	GND
12	+3.3V	TRDY#	GND	DEVSEL#
13	FRAM1#	GND	IRDY#	+3.3V
14	GND	AD16	+3.3V	C/BE2#
15	AD18	+3.3V	AD17	GND
16	AD21	AD20	GND	AD19
17	+3.3V	AD23	AD22	+3.3V
18	IDSEL0	GND	IDSEL1	IDSEL2
19	AD24	C/BE3#	V/I/O	IDSEL3
20	GND	AD26	AD25	GND
21	AD29	+5V	AD28	AD27
22	+5V	AD30	GND	AD31
23	REQ0#	GND	REQ1#	V/I/O
24	GND	REQ2#	+5V	GNT0#
25	GNT1#	V/I/O	GNT2#	GND
26	+5V	CLK0	GND	CLK1
27	CLK2	+5V	CLK3	GND
28	GND	INTD#	+5V	RST#
29	+12V	INTA#	INTB#	INTC#
30	-12V	REQ3#	GNT3#	GND

J2/P2		
Pin	Row D	Row C
0	GND	GND
1	MEMCS16*	SBHE*
2	IOCS16*	LA23
3	IRQ10	LA22
4	IRQ11	LA21
5	IRQ12	LA20
6	IRQ15	LA19
7	IRQ14	LA18
8	DACK0*	LA17
9	DRQ0	MEMR*
10	DACK5*	MEMW*
11	DRQ5	SD8
12	DACK6*	SD9
13	DRQ6	SD10
14	DACK7*	SD11
15	DRQ7	SD12
16	+5V	SD13
17	MASTER*	SD14
18	GND	SD15
19	GND	KEY

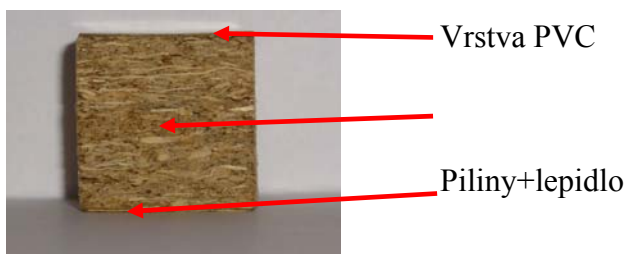
J1/P1		
Pin	Row A	Row B
1	IOCHK*	GND
2	SD7	RESET
3	SD6	+5V
4	SD5	IRQ9
5	SD4	-5V
6	SD3	DRQ2
7	SD2	-12V
8	SD1	SRDY*
9	SD0	+12V
10	IOCHRDY	KEY
11	AEN	SMEMW*
12	SA19	SMEMR*
13	SA18	IOW*
14	SA17	IOR*
15	SA16	DACK3*
16	SA15	DRQ3
17	SA14	DACK1*
18	SA13	DRQ1
19	SA12	REFRESH*
20	SA11	BCLK
21	SA10	IRQ7
22	SA9	IRQ6
23	SA8	IRQ5
24	SA7	IRQ4
25	SA6	IRQ3
26	SA5	DACK2*
27	SA4	TC
28	SA3	BALE
29	SA2	+5V
30	SA1	OSC
31	SA0	GND
32	GND	GND

PŘÍLOHA P V: ZÁKLADNÍ INFORMACE O DŘEVOTŘÍSCE

Jedná se o materiál, který je vytvořen ze zbytků dřevěných pilin tak, že se tyto piliny spojí nějakým průmyslovým pojidlem, a slisují se za velkého tlaku do tvaru desky. Jelikož se tento materiál zejména používá ve stolařské a nábytkářské výrobě, bývá potažen vrstvou umělé hmoty, která může mít různý vzhled a tepelně izolační vlastnosti. Uvažujeme-li desku o šířce 1,9 cm, pak může taková deska mít tento průřez:



Vlastnosti tohoto materiálu nejsou obecně přesně dané - musejí se ověřovat experimentálně pro každou jednu desku, protože každý výrobce používá k výrobě dřevotřískových desek jiných pilin a jiných pojidel. Obecně však mívá nižší hustotu než dřevo, ze kterého byla vyrobena, a bývá dobrý izolant, neboť ve své struktuře obsahuje hodně vzduchu. Tepelná kapacita bývá obvykle dobrá. Pevnost bývá dobrá v tahu a tlaku, ovšem náhlá velká změna síly působící na její povrch může způsobit ztrátu integrity spojeného materiálu a dojde ke zlomení – bývá tedy křehká.



Obrázek dřevotřísky



Obrázek smrkového dřeva

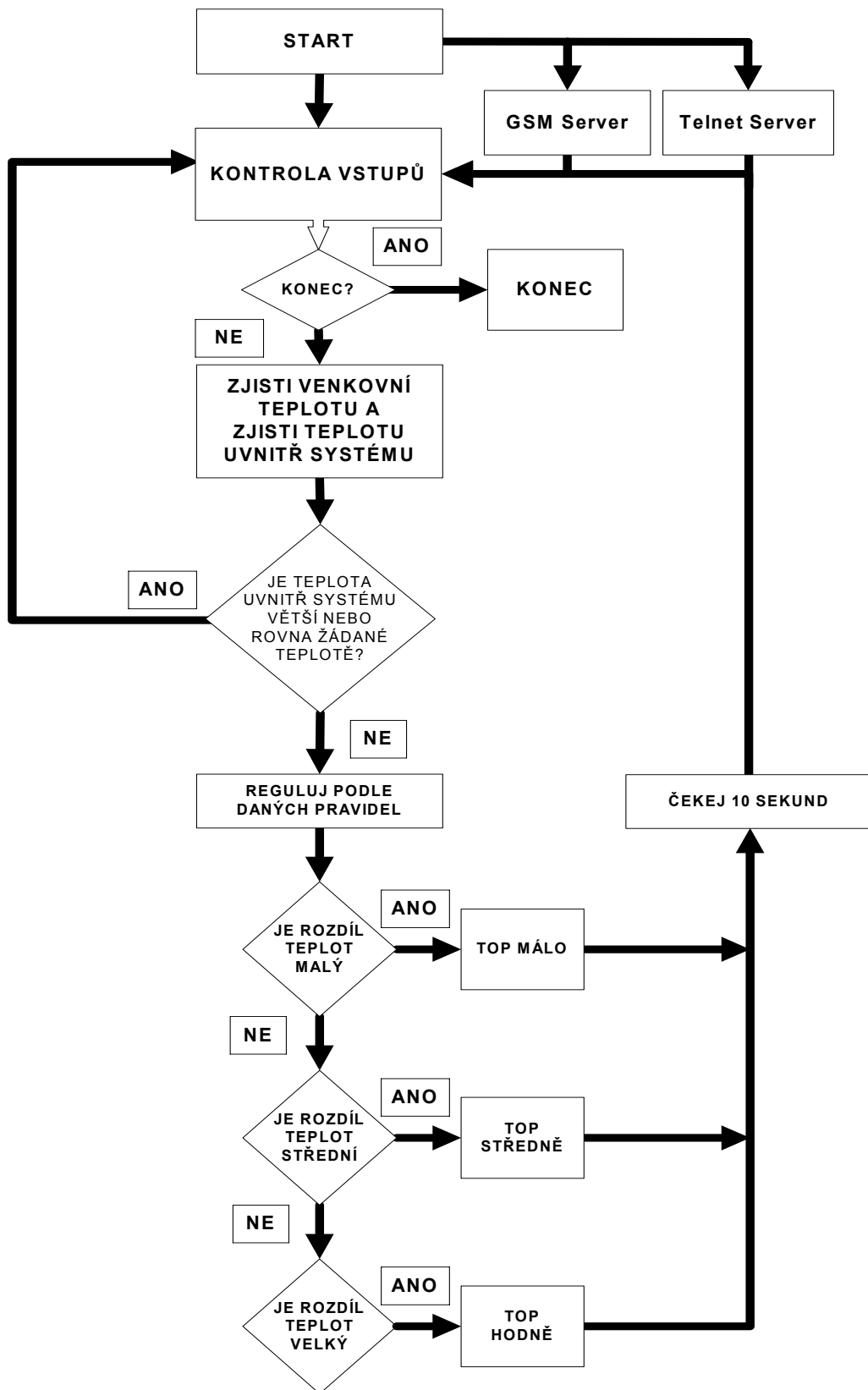
PŘÍLOHA P VI: SEZNAM PŘÍKAZŮ POUŽITELNÝCH PRO GSM SERVER

Číslo	SMS příkaz + syntaxe	Význam
1.	<p>„GET-TEMP<i>kde</i>”</p> <p><i>kde</i> = „-IN” nebo „-OUT”</p> <p>Př.: „GET-TEMP-IN”, „GET-TEMP-OUT”</p>	Požadavek na posláni aktuální informace o teplotě soustavy . Teplota může být vnitřní nebo vnější.
2.	<p>„GET-REG-TEMP“ (bez parametru)</p>	Požadavek na posláni aktuální nastavené žádané teplotě regulované soustavy.
3.	<p>„SET-REG-TEMP<i>teplota</i>“</p> <p><i>teplota</i>... “=XX.X C”</p> <p>Př.: „SET-REG-TEMP=28.3 C”</p>	Požadavek na změnu aktuální žádané teploty v regulované soustavě na teplotu XX.X °C
4.	<p>„START-REG<i>čas</i>“</p> <p><i>čas</i>... „=yy/mm/dd hh:mi:ss“, “-NOW”</p> <p>Př.: „START-REG=060919 21201” nebo “START-REG-NOW”</p>	Požadavek na zahájení regulace , čas musí být roven nebo větší než aktuální čas v Biscuit PC a ve formátu „yy/mm/dd hh:mi:ss”.
5.	<p>„STOP-REG<i>čas</i>“</p> <p><i>čas</i>... „=yy/mm/dd hh:mi:ss“, “-NOW”</p> <p>Př.: „STOP-REG=060919 100005” nebo “STOP-REG-NOW”</p>	Požadavek na ukončení regulace , čas musí být roven nebo větší než aktuální čas v Biscuit PC a ve formátu „yy/mm/dd hh:mi:ss”.
6.	<p>„SCHEDULE-REG:<i>čas</i>; <i>žádaná teplota</i>;“</p> <p><i>čas</i>... “=yy/mm/dd hh:mi:ss”</p> <p><i>ž.teplota</i>... “XX.X C”</p> <p>Př.: “SCHEDULE-REG=06/01/01 12:02:11;30.0 C;”</p>	Požadavek o zavedení plánovaného startu regulace , a to na čas „yy/mm/dd hh:mi:ss”, na žádanou teplotu XX.X °C. Tato funkce je experimentální (ve stádiu vývoje).
7.	<p>„QUIT-REG“ (bez parametru)</p>	Požadavek na ukončení programu regulátor .

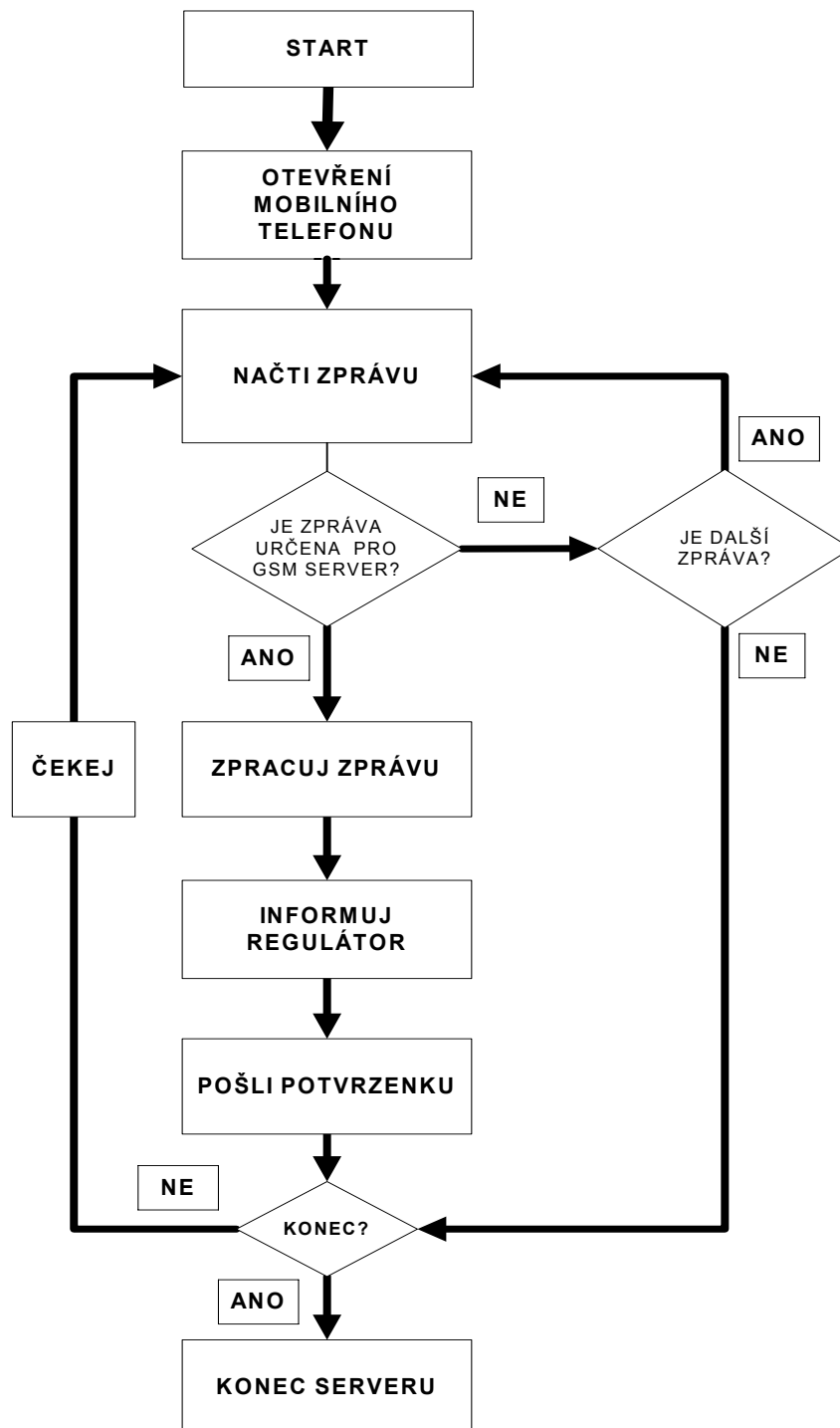
PŘÍLOHA P VII: SEZNAM PŘÍKAZŮ POUŽITELNÝCH PRO TELNET SERVER

Číslo	SMS příkaz + syntaxe	Význam
1.	„HELP“ (bez parametru)	Příkaz vypíše nápovědu a seznam příkazů pro tento server.
2.	„GET-STATUS“ (bez parametru)	Příkaz vypíše stav regulátoru k dané chvíli.
3.	„GET-TEMPkde” kde= „-IN” nebo „-OUT” Př.: „GET-TEMP-IN”, „GET-TEMP-OUT”	Požadavek na posláni aktuální informace o teplotě soustavy . Teplota může být vnitřní „-IN“ nebo vnější „-OUT“.
4.	„GET-REG-TEMP“ (bez parametru)	Požadavek na posláni aktuální nastavené žádané teplotě regulované soustavy.
5.	„SET-REG-TEMPteplota” teplota... “=XX.X C” Př.: „SET-REG-TEMP=28.3 C”	Požadavek na změnu aktuální žádané teploty v regulované soustavě na teplotu XX.X °C
6.	„START-REGčas” čas... „=yy/mm/dd hh:mi:ss”, “-NOW” Př.: „START-REG=060919 21201” nebo “START-REG-NOW”	Požadavek na zahájení regulace , čas musí být roven nebo větší než aktuální čas v Biscuit PC a vyhovovat formátu „yy/mm/dd hh:mi:ss”.
7.	„STOP-REGčas” čas... „=yy/mm/dd hh:mi:ss”, “-NOW” Př.: „STOP-REG=060919 10005” nebo “STOP-REG-NOW”	Požadavek na ukončení regulace , čas musí být roven nebo větší než aktuální čas v Biscuit PC s vyhovovat formátu čas „yy/mm/dd hh:mi:ss”
8.	„SCHEDULE-REG:čas; žádaná teplota;” čas... “=yy/mm/dd hh:mi:ss” ž.teplota... “XX.X C” Př.: “SCHEDULE-REG=06/01/01 12:02:11;30.0 C;”	Požadavek o zavedení plánovaného startu regulace , a to na čas „yy/mm/dd hh:mi:ss”, na žádanou teplotu XX.X °C. Tato funkce je experimentální (ve stádiu vývoje).
9.	„QUIT-REG” (bez parametru)	Požadavek na ukončení programu regulátor .

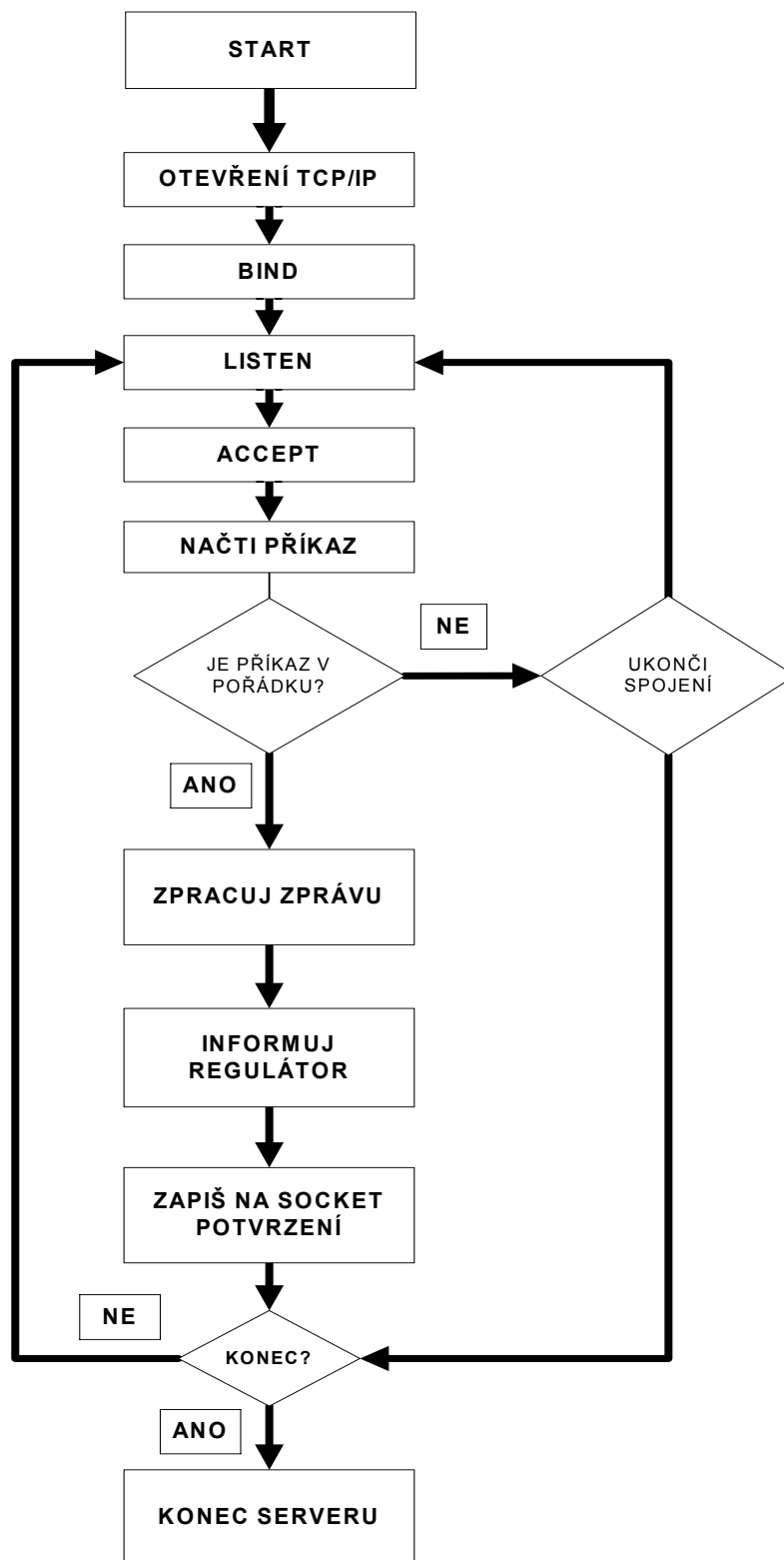
PŘÍLOHA P VIII: BLOKOVÉ SCHÉMA FUZZY REGULÁTORU



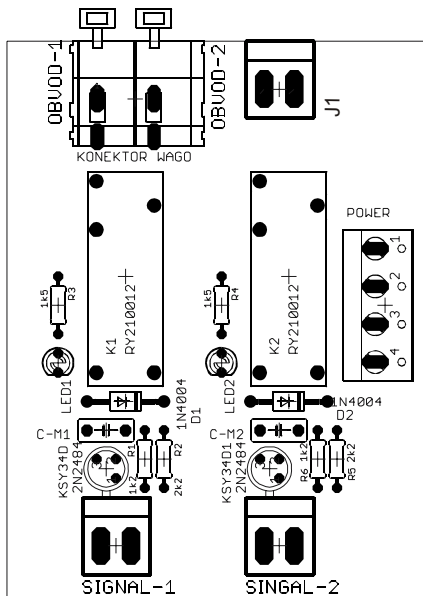
PŘÍLOHA P IX: BLOKOVÉ SCHÉMA GSM SERVERU



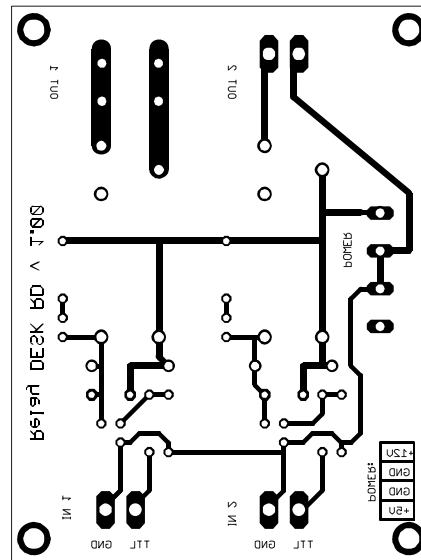
PŘÍLOHA P X: BLOKOVÉ SCHÉMA TELNET SERVERU



PŘÍLOHA P XI: VZHLED NÁVRHU DESKY RELAY DESK V1.0, SCHEMA ZAPOJENÍ.



Vzhled desky ze strany mědi -
zobrazeno rozložení součástek.



Vzhled desky ze strany mědi -
zobrazen plošný spoj.

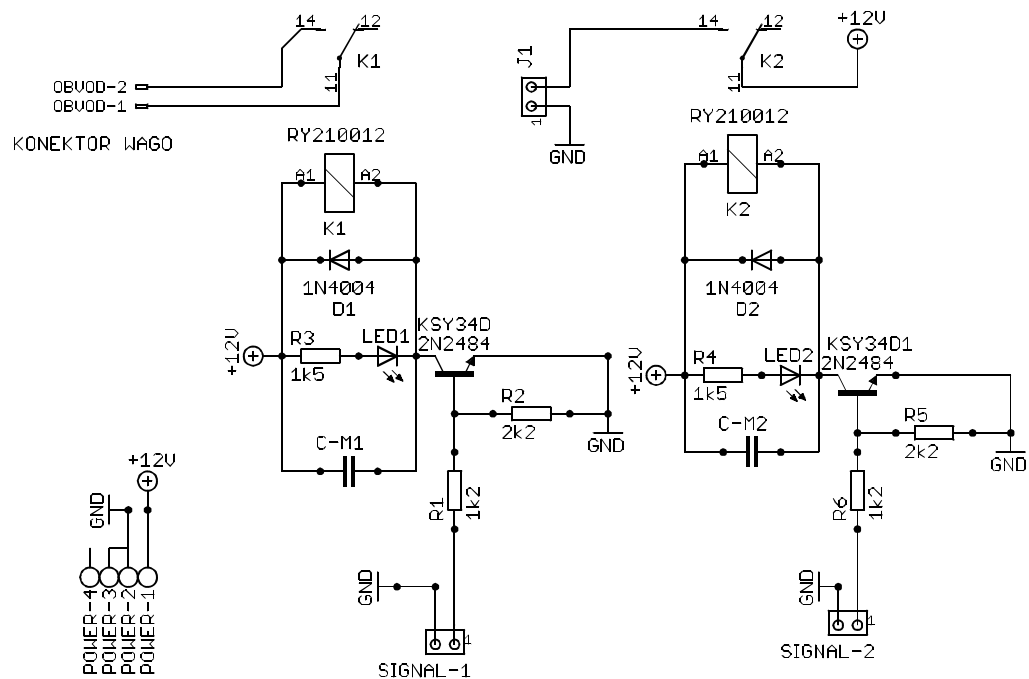
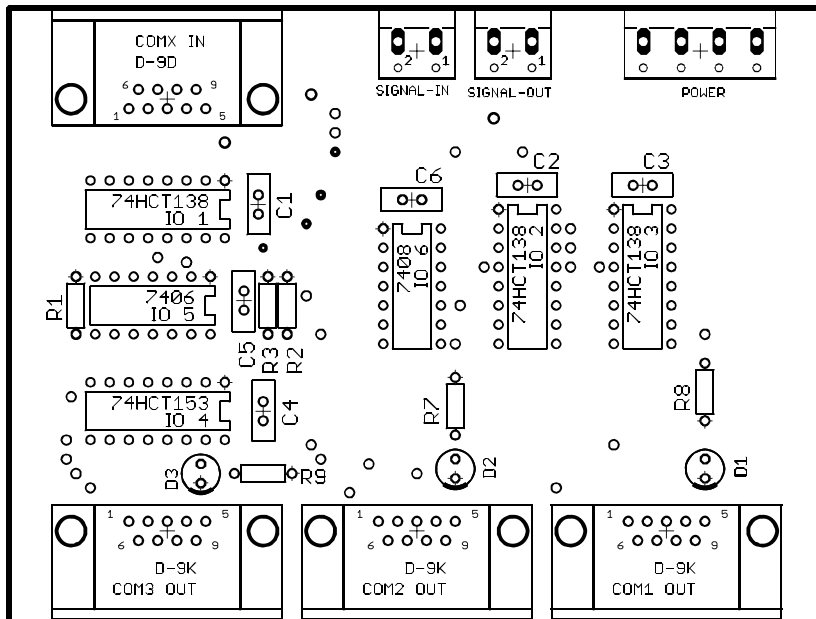


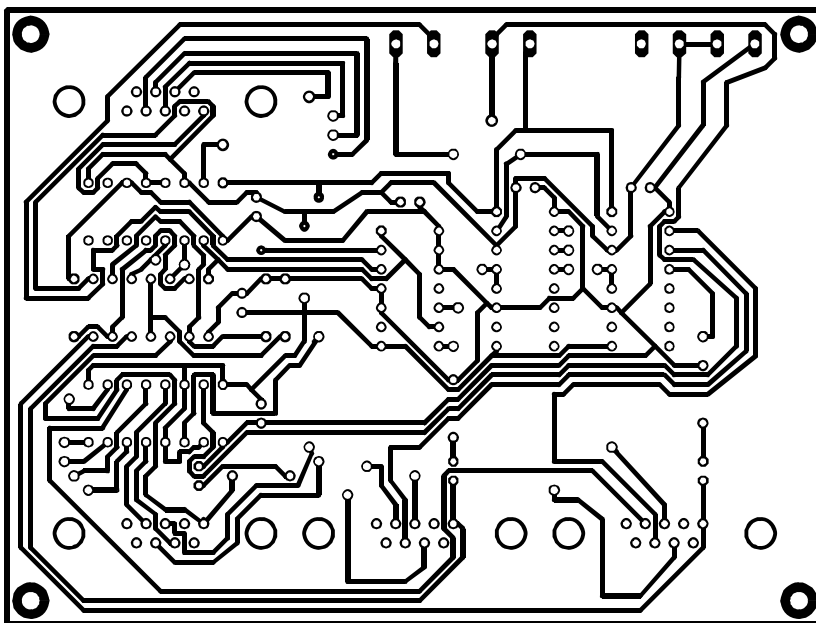
Schéma zapojení jednotlivých součástí.

PŘÍLOHA P XIII: VZHLED NÁVRHU DESKY COM SWITCH V1.0

a) Vzhled desky s rozložením součástek

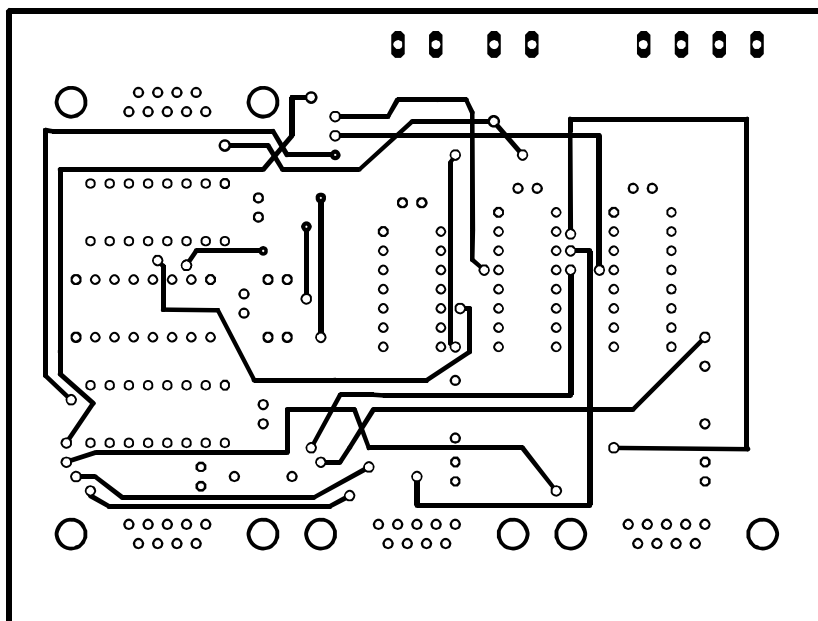


b) Vzhled desky ze strany mědi se zobrazením plošných spojů



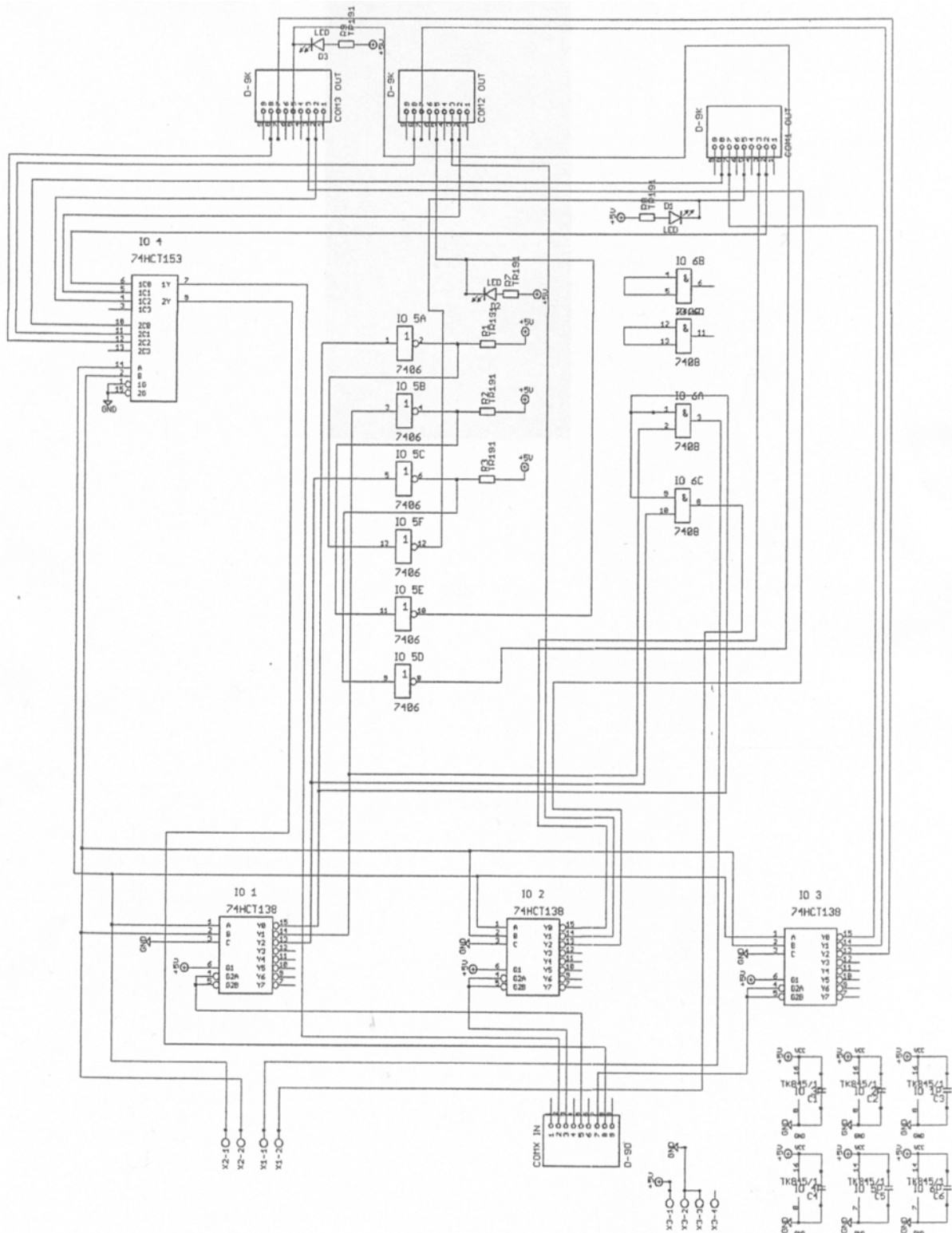
PŘÍLOHA P XIIB: VZHLED NÁVRHU DESKY COM SWITCH V1.0

a) Vzhled desky ze strany mědi s rozložením vzdušných spojů



PŘÍLOHA P XIII: VZHLED NÁVRHU DESKY COM SWITCH V1.0

c) Schéma zapojení součástek



Pozn.: Toto je vzhled celého schématu pro desku Com Switch v1.0. Schéma bylo spojeno ze čtyř částí do jedné.