

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

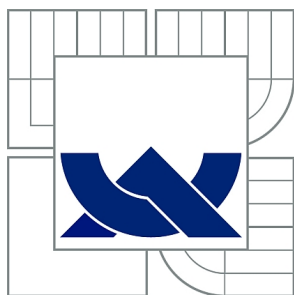
SYSTÉM NA MONITOROVÁNÍ TEPLoty

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

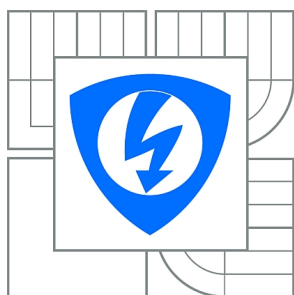
AUTOR PRÁCE
AUTHOR

MILAN PAPEŽ

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNologiÍ**
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SYSTÉM NA MONITOROVÁNÍ TEPLoty

SYSTEM FOR TEMPERATURE MONITORING IN SERVER ROOMS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

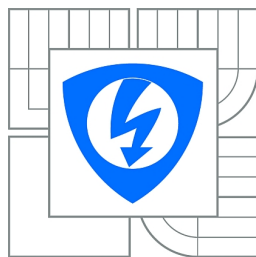
AUTOR PRÁCE
AUTHOR

MILAN PAPEŽ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ MACHO, Ph.D.

BRNO 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Milan Papež

ID: 120609

Ročník: 3

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Systém na monitorování teploty

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s problematikou monitorování teploty v serverovnách a rozvodných uzlech.
2. Navrhněte koncepci zařízení, které by sledovalo teplotu v serverovně a při překročení nastavené hodnoty by odeslalo e-mail.
3. Vyberte vhodný mikrokontrolér a snímače teploty. Nakreslete blokové schéma zařízení.
4. Navrhněte obvodové řešení zařízení, vypočtěte hodnoty jednotlivých součástí, navrhněte plošné spoje. Vytvořte výrobní dokumentaci.
5. Pro zařízení vytvořte softwarové vybavení a odlaďte ho.

DOPORUČENÁ LITERATURA:

[1] KREIDL, Marcel. MĚŘENÍ TEPLITY - senzory a měřicí obvody. 1. vyd. Praha: BEN, 2005. 240 s. ISBN 80-7300-145-4.

Termín zadání: 7.2.2011

Termín odevzdání: 30.5.2011

Vedoucí práce: Ing. Tomáš Macho, Ph.D.

prof. Ing. Pavel Jura, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Cílem této práce je návrh a realizace systému na sledování teploty v serverovně, který je schopen odeslat e-mail při překročení nastavených hodnot teploty. Pro řešení tohoto úkolu bylo využito mikrokontroléru PIC18F97J60, který obsahuje periférii ethernetového řadiče. Vlastní odeslání e-mailu je řešeno SMTP protokolem, který je obsažen v TCP/IP stacku poskytovaného výrobcem mikrokontroléru. Měření teploty je uskutečněno využitím snímačů teploty DS18B20, které komunikují přes rozhraní 1-Wire. Kromě schopnosti odeslat e-mail umožňuje zařízení také jeho vzdálenou konfiguraci. Hlídání teploty bylo testováno s využitím pěti snímačů.

KLÍČOVÁ SLOVA

System na monitorování teploty, e-mail, TCP/IP stack, SMTP, 1-Wire,

ABSTRACT

The aim of this thesis is the design and implementation of a system for temperature monitoring in a server room that is able to send e-mail when exceeding the set temperature value. For this purpose was used PIC18F97J60 microcontroller that includes an Ethernet controller peripheral. Sending e-mail is realized via SMTP protocol, which is included in the TCP / IP stack provided by the manufacturer of the microcontroller. For temperature measurement is used DS18B20 temperature sensors that communicate via the 1-Wire interface. Device also allows the remote configuration. Temperature monitoring was tested using five sensors.

KEYWORDS

System for temperature monitoring in server rooms, e-mail, TCP/IP stack, SMTP, 1-Wire,

PAPEŽ, Milan *System na monitorování teploty*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2011. 70 s. Vedoucí práce byl Ing. Tomáš Macho, PhD.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Systém na monitorování teploty“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu Ing.Tomáši Machovi, PhD. za užitečné rady a připomínky poskytnuté během plnění cílů bakalářské práce.

Brno

.....

(podpis autora)

OBSAH

1	Úvod	12
2	Monitorování teploty v serverovných a rozvodných uzlech	13
2.1	Fyzická infrastruktura	13
2.2	Rozložení teploty	14
2.3	Požadavky kladené na návrh zařízení	15
3	Návrh koncepce zařízení	16
3.1	Blokové schéma zařízení	18
3.2	Mikrokontrolér	19
3.2.1	Ethernetový modul	19
3.2.2	I2C modul	21
3.3	Snímač teploty	22
3.3.1	1-Wire	22
3.3.2	DS18B20	25
3.3.3	DS2482-100	28
4	Návrh zařízení	31
4.1	Obvodový návrh	31
4.1.1	Zapojení mikrokontroléru	31
4.1.2	Připojení snímačů DS18S20	33
4.1.3	Ethernetový transformátor	34
4.1.4	Napájení	35
4.2	Návrh plošného spoje	36
4.2.1	Číslicová část	36
4.2.2	Napájecí část	37
5	Softwarové vybavení	38
5.1	Hlavní program	39
5.1.1	Konfigurace stacku	39
5.1.2	Vytvoření aplikace s Microchip TCP/IP stack	40
5.1.3	Popis hlavního programu	41
5.2	Komunikace se snímači teploty	43
5.2.1	Rutiny pro rozhraní I2C a 1-Wire	43
5.2.2	Vyhledávání a adresace 1-Wire zařízení	45
5.3	Odesílání emailu při překročení určených mezí teploty	49
5.3.1	Odeslání emailu a SMTP protokol	49

5.3.2	Algoritmus pro periodické měření teploty a odeslání emailu při překročení povolených mezí teplot	51
5.4	Příkazy pro konfiguraci přístroje a měření teploty	53
5.4.1	Úloha pro zpracování příkazů	53
5.4.2	Ukázka komunikace s přístrojem	55
6	Závěr	59
	Literatura	60
	Seznam symbolů, veličin a zkratk	62
	Seznam příloh	63
A	Výrobní dokumentace	64
A.1	Schema zapojení	64
A.2	Motivy plošných spojů	65
A.3	Osazovací výkresy	66
A.4	Motiv padů	67
A.5	Seznam součástek	68
A.6	Fotografie výrobku	70

SEZNAM OBRÁZKŮ

2.1	Místnost datového centra s vyznačením proudění vzduchu	13
2.2	Termální zobrazení teploty vzduchu v horké uličce	14
3.1	Propojení mikrokontroléru a ENC28J60	16
3.2	Struktura RTCS TCP/IP stacku	17
3.3	Blokové schéma systému na monitorování teploty	18
3.4	Rámec podle IEEE 802.3	19
3.5	Ethernetový modul mikrokontroléru PIC16F97J60	20
3.6	Microchip TCP/IP stack	21
3.7	Část přenosu na I2C	22
3.8	Resetovací puls	23
3.9	Zápis jednoho bitu	23
3.10	Čtení jednoho bitu	24
3.11	Blokové schéma snímače teploty DS18B20	25
3.12	ROM kód DS18B20	25
3.13	Soubor registrů DS18B20	26
3.14	Blokové schéma DS2482-100	28
3.15	Konfigurační registr DS2482-100	28
3.16	Stavový registr DS2482-100	29
4.1	Zapojení mikrokontroléru a převodníku DS2482-100	32
4.2	Zapojení ethernetového transformátoru	34
4.3	Zapojení napájecí části	35
4.4	Deska plošného spoje v oblasti procesoru	36
4.5	Deska plošného spoje v oblasti napájení	37
5.1	Hierarchie modulů softwarového vybavení mikrokontroléru	38
5.2	Hlavní program	41
5.3	I2C sekvence pro vykonání restartu, zápisu a čtení na 1-Wire rozhraní	44
5.4	Vývojový diagram pro vyhledávání 1-Wire zařízení za pomoci příkazu 1-Wire Triplet	48
5.5	Posloupnost operací při odesílání emailu	49
5.6	Vývojový diagram pro periodické měření teploty a odesílání emailu	52
5.7	Stavový diagram úlohy pro zpracování příkazů	54
5.8	Ukázka komunikace s přístrojem přes program HyperTerminál	55
A.1	Schema zapojení	64
A.2	Motiv spojů horní strany desky	65
A.3	Motiv spojů spodní strany desky	65
A.4	Osazovací výkres horní strany desky	66
A.5	Osazovací výkres spodní strany desky	66

A.6	Motiv padů desky	67
A.7	Výsledný výrobek	70

SEZNAM TABULEK

5.1	Možné stavy na 1-Wire sběrnici při vyhledávání ROM kódu	46
5.2	Příkazy pro konfiguraci přístroje - 1. část	56
5.3	Příkazy pro konfiguraci přístroje - 2. část	57
5.4	Příkazy pro měření teploty	58

1 ÚVOD

V dnešní době se internet stal již neodmyslitelnou součástí života mnohých z nás. Trend internetu neustále roste a je tak třeba zřizovat mnoho nových rozvodných uzlů a serveroven, aby byla zajištěna dostupnost a kvalita komunikace ve všech částech světa. Pro správnou funkci prostředků obstarávajících chod internetu je důležité zajistit vhodné podmínky provozu. Jednou z těchto podmínek je teplota. V prostorech, kde je soustředěno velké množství zařízení, která jsou charakteristická vysokým výpočetním výkonem, což je typické pro serverovny a rozvodné uzly, dochází k velké produkci ztrátového tepla. Proto jsou tyto prostory vybaveny systémem klimatizace, který drží jejich teplotu pokud možno na optimální úrovni. Tyto systémy mohou být sice vybaveny diagnostikou, která dokáže nahlásit jejich případnou poruchu, to ale nemusí být vždy pravda a navíc se může stát, že i tato část zařízení selže. Proto bývají výše uvedené prostory vybaveny systémem na monitorování teploty, jehož návrhem se tato práce zabývá.

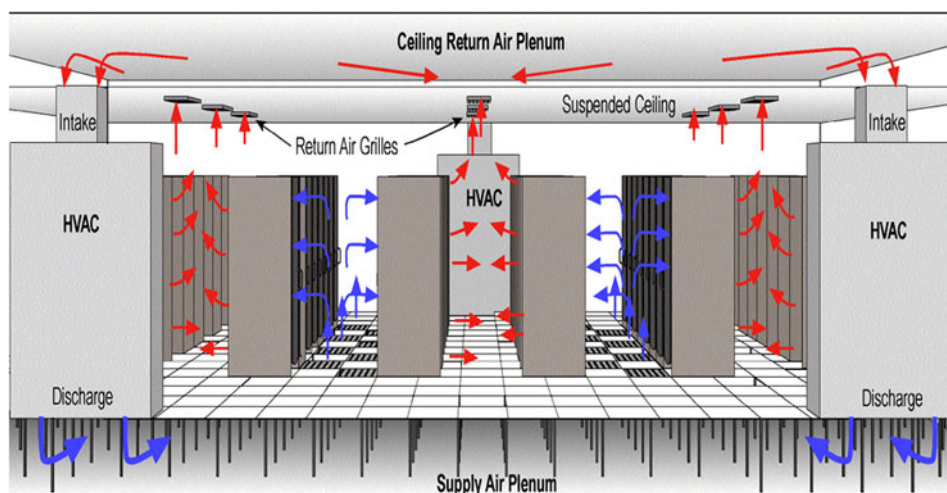
2 MONITOROVÁNÍ TEPLoty V SERVEROVNÁCH A ROZVODNÝCH UZLECH

Fyzická infrastruktura a rozložení teploty jsou důležitým faktorem, který je nutno zvážit před návrhem koncepce zařízení pro monitorování teploty. Proto bude dále uveden krátký rozbor této problematiky.

Struktura serveroven a rozvodných uzlů může být velice různorodá. Většinou se ale jedná o speciálně vytyčené prostory, ve kterých se nachází zařízení realizující činnost typickou jak pro rozvodné uzly (switche, routery), tak serverovny (servery) dohromady. V následujícím budou tyto prostory označovány jako datová centra.

2.1 Fyzická infrastruktura

Fyzická infrastruktura datových center je různá podle velikosti společnosti, ve které se datové centrum nachází. Obecně je datové centrum realizováno modulárně pomocí tzv. rackových stojanů, do kterých jsou vkládány jednotlivé komponenty zajišťující chod centra. Může se jednat o malé místnosti s jedním stojanem (typické například pro firemní LAN servery) nebo o velké haly, kde jsou až stovky stojanů (velké IT společnosti).



Obr. 2.1: Místnost datového centra s vyznačením proudění vzduchu [8]

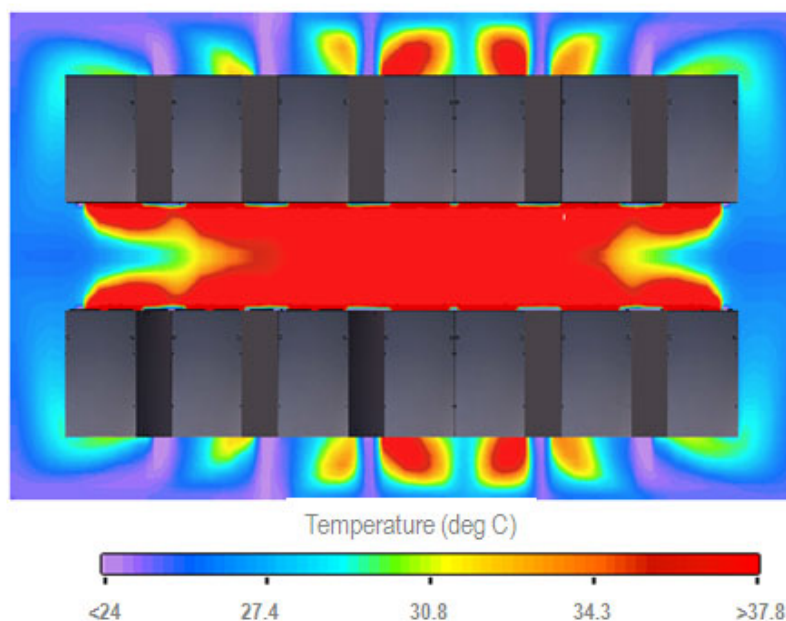
Zaměříme-li se na datová centra střední velikosti, tak jak ukazuje obrázek 2.1. Můžeme si povšimnout, že rackové stojany jsou zde postaveny na zdvojené podlaze, která slouží pro rozvádění chladného vzduchu po místnosti. Dále jsou stojany uspořádávány do systémů tzv. chladných a horkých uliček podle smyslu proudění vzduchu

vždy chladnými a horkými stranami k sobě. Chladné uličky bývají mnohdy zakryty ze všech stran, aby nedocházelo k mísení studeného a horkého vzduchu a dosáhlo se zároveň i větší efektivity chlazení. Nad každou horkou uličkou se potom nachází sání klimatizace, které odvádí horký vzduch. Existují řešení, kde jsou uzavřeny i horké uličky, zde je horký vzduch odváděn do sání klimatizace přímo.

2.2 Rozložení teploty

Jednotlivá zařízení datového centra jako například servery se obvykle vyznačují velkým výpočetním výkonem, proto je jasné, že při jejich činnosti bude docházet ke značnému uvolňování ztrátového tepla. Takto uvolňované teplo by se bez působení klimatizace v místnosti datového centra hromadilo, až by způsobilo automatické odstavení jednotlivých zařízení.

Působení klimatizace, fyzická infrastruktura a to že každý stojan může být jednotlivými komponentami osazen v různé míře, nás přivádí k tomu, že rozložení teploty v místnosti je nehomogenní. O tomto faktu vypovídá i obrázek 2.2, kde je vidět zobrazení teploty vzduchu v horké uličce.



Obr. 2.2: Termální zobrazení teploty vzduchu v horké uličce [7]

Průměrná teplota v datovém centru se pohybuje v rozmezí 20 až 25°C. Konkrétně ve zdvojené podlaze tato hodnota činí 16 až 19°C, dále v místnosti 18 až 24°C a 24 až 40°C v rackovém stojanu [9].

2.3 Požadavky kladené na návrh zařízení

V předchozím rozboru bylo ukázáno, že rozložení teploty v místnosti datového centra je nehomogenní. Z tohoto faktu vyplývá, že je třeba sestavit systém, který by umožnil měření teploty na několika místech. Pouze jeden snímač, byť sebelépe umístěný, by nestačil včasné reagovat na změnu teploty rackového stojanu vzdáleného o jeden nebo dva metry dále. Zde tedy vyvstává první požadavek, a to je potřeba snímačů teploty, které umožňují komunikovat se systémem na větší vzdálenost. Tyto snímače musí měřit rozsah teploty alespoň 10 až 50°C.

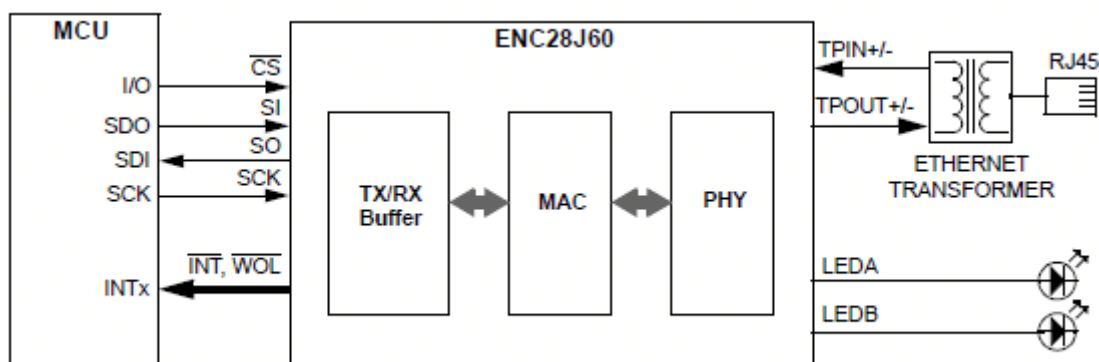
Další požadavek, tedy potřeba odesílat email při překročení povoleného rozsahu teplot, je určen ze zadání. Zde se nastiňují dvě možnosti, jak koncipovat zařízení v tomto směru. První možností je, že se zařízení navrhne jako embedded systém obsahující mikrokontrolér vybavený rozhraním USB nebo RS232. Pomocí některého z těchto rozhraní by mohl být připojen k jednomu ze serverů datového centra. Na tomto serveru by potom běžel obslužný software, který by zajistil odeslání emailu. Druhou možností je zvolit takovou hardwarovou koncepci, která by umožnila přímé připojení do sítě ethernet. To může být řešeno mikrokontrolérem a externím radičem mikrokontroléru nebo mikrokontrolérem, který má tento radič vestavěný přímo v sobě. Do mikrokontroléru je potom třeba nahrát TCP/IP stack obsahující SMTP protokol, aby bylo možno odeslat email.

3 NÁVRH KONCEPCE ZAŘÍZENÍ

V závěru předchozí kapitoly byly uvedeny požadavky, které by měl navrhovaný systém splňovat. Byly také navrženy dvě možnosti způsobu odeslání emailu. Druhá z možností, která není závislá na provozu jednoho ze serverů v místnosti datového centra se jeví jako lepší řešení. Hlavním kritériem pro úspěšný návrh systému se tak stává zvolení takové hardwarové koncepce, která by zahrnovala jednoduchou implementaci již vytvořeného TCP/IP stacku. Závěr předchozí kapitoly také nastínil dva možné pohledy na volbu této hardwarové koncepce. Ještě před popisem finální koncepce následující text popíše obě tyto možnosti na příkladech řešení, které byly při návrhu uvažovány. Dále bude uvedeno blokové schéma navržené koncepce, kde budou také uvedeny důvody pro volbu použitého mikrokontroléru a snímačů teploty. Hlavní komponenty zapojení budou v samostatných podkapitolách krátce popsány.

ENC28J60

ENC28J60 je externí řadič ethernetu implementující v sobě funkce fyzické a linkové vrstvy komunikačního modelu ISO/OSI. Pro svojí činnost vyžaduje krystalový oscilátor o kmitočtu 25MHz, ethernetový transformátor a řídicí mikrokontrolér obsahující periférii SPI. TCP/IP stack by byl implementován ve spolupracujícím mikrokontroléru a vyžadoval by soubor rutin pro komunikaci s tímto obvodem. Rychlost přenosu je zde pouze ve verzi 10Mbps. Možné blokové schéma tohoto řešení je na obrázku 3.1.



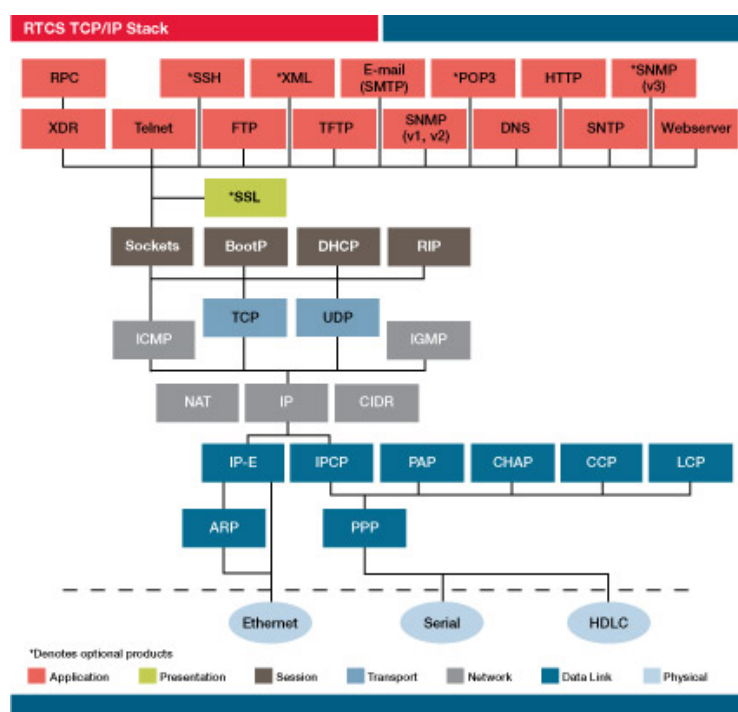
Obr. 3.1: Propojení mikrokontroléru a ENC28J60 [11]

Jako vhodný TCP/IP stack může být použit tzv. uIP TCP/IP stack. Tento je navržen speciálně pro 8 bitové mikrokontroléry. Neklade velké nároky na paměť programu a paměť RAM, protože realizace jednotlivých vrstev obsahuje pouze to

nejnutnější. Tento poskytuje protokoly ARP, SLIP, IP, UDP, ICMP a TCP. Dále jsou pro něj přichystány i aplikační protokoly jako web server, web client, SMTP client, Telnet server a DNS.

MCF52232

MCF52232 je 32 bitový Coldfire V2 mikrokontrolér s redukovanou instrukční sadou, který vyrábí firma Freescale. Tento mikrokontrolér je vybaven bohatou sadou periférií, mezi které patří tzv. FEC (Fast Ethernet Controller) a EPHY (Ethernet Physical Transceiver). Pomocí těchto periférií je možno provést přímé připojení mikrokontroléru na ethernet. Rychlost komunikace po ethernetu je zde podporována ve verzi 10Mbps a 100Mbps.

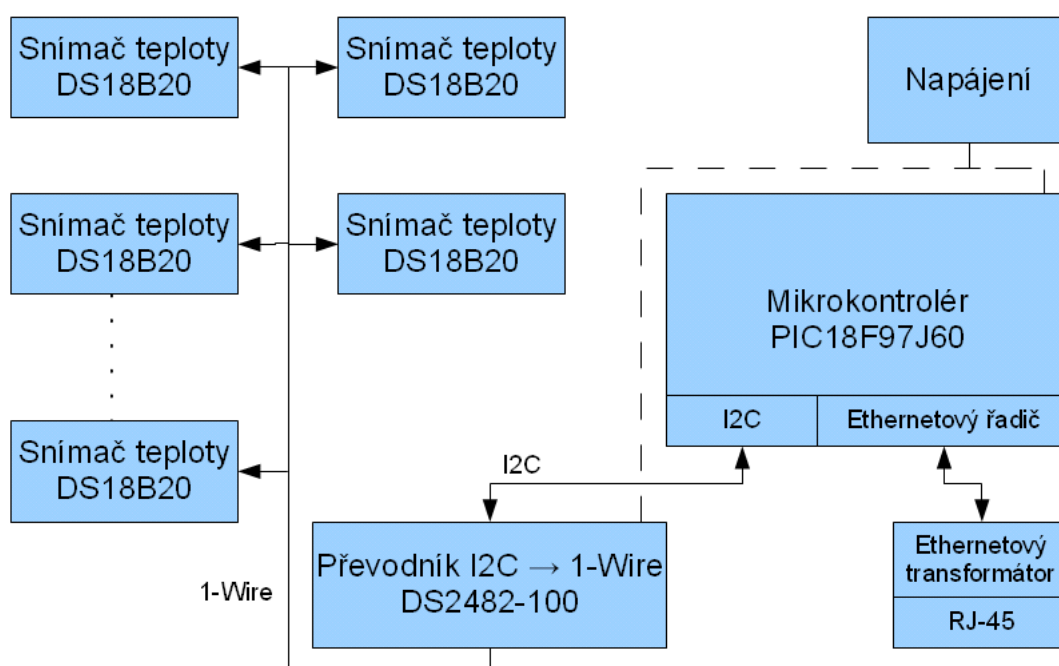


Obr. 3.2: Struktura RTCS TCP/IP stacku [12]

Firma Freescale poskytuje k tomuto mikrokontroleru TCP/IP stack RTCS (Real-Time TCP/IP Communication Suite), který je zahrnut v real-time operačním systému MQX. Systém pro monitorování teploty v datovém centru nepatří mezi časově kritické aplikace a proto není třeba, aby běžel pod operačním systémem reálného času. Výčet poskytovaných protokolů je značný, proto není vypsán, ale uveden na obrázku 3.2.

3.1 Blokové schéma zařízení

Dříve uvedená možnost využití ENC28J60 poukázala na to, že firma Microchip implementuje tento druh řadiče do některých svých mikrokontrolérů. Navíc bylo zjištěno, že k těmto mikrokontrolérům nabízí tato firma také vlastní TCP/IP stack (Microchip TCP/IP stack). Využitím mikrokontroléru PIC18F97J60 s tímto vestavěným řadičem ethernetu se zjednoduší návrh plošného spoje. Ze softwarového hlediska se v porovnání s dříve uvedenými možnostmi zdá být tato volba kompromisem. Vůči první je tato možnost robustnější, protože TCP/IP stack od firmy Microchip není napsán jen v té nejnutnější podobě a vůči druhé v jednoduchosti, protože pronikání do operačního systému MQX by zabralo více času.



Obr. 3.3: Blokové schéma systému na monitorování teploty

Při výběru vhodného snímače byl jako první uvažován MCP9801 vyráběný firmou Microchip. Ten využívá pro komunikaci s mikrokontrolérem rozhraní I2C. Vzhledem k tomu, že je třeba snímače připojit na větší vzdálenost, je důležité přihlédnout ke kapacitě vodiče. Standard I2C povoluje maximální kapacitu sběrnice o hodnotě 400pF. Při použití vodiče, který by měl kapacitu přibližně 50pF/m, by celková vzdálenost na natažení snímačů byla poměrně malá (připočteme-li kapacitu cesty na plošném spoji, kapacitu kontaktů a kapacitu přívodů snímače a procesoru). Bylo by sice možné použít opakovačů P82B96 firmy NXP Semiconductors, ale toto řešení by pouze navyšovalo cenu zařízení. Aby bylo možno realizovat připojení většího počtu snímačů teploty na větší vzdálenosti, byl vybrán teplotní snímač

DS18B20, který komunikuje pomocí jednovodičové sběrnice Dallas (dále jen 1-Wire). Rozhraní 1-Wire dovoluje natáhnout vedení až na stovky metrů.

Celé zařízení je koncipováno jako embedded systém, u kterého se nepočítá s tím, že by ho musel uživatel obsluhovat přímo fyzicky. Není tedy třeba, aby obsahoval jakékoliv ovládací (fyzické ovládací) a zobrazovací prvky. To z toho důvodu, že do místnosti datového centra (serverovny) není přístup příliš častý. Možnost jeho ovládání bude řešena prostřednictvím TCP spojení. Zobrazovací a ovládací komponenty by tedy pouze navyšovali cenu zařízení. Blokové schéma navržené koncepce řešení ukazuje obrázek 3.3.

3.2 Mikrokontrolér

Mikrokontrolér PIC18F97J60 je 8-bitový mikrokontrolér s redukovanou instrukční sadou, který v sobě zapouzdřuje ethernetový modul. Obsahuje 128KB paměti programu a 3808B paměti RAM. Rozsah napájecího napětí je 3,1V až 3,6V. Maximálně podporovaný vnitřní kmitočet je 41.667MHz. Je zde sada standardních periférií mezi něž patří USART, SPI, I2C, ADC, Capture/Compare/PWM modul atd. Pro tuto práci je nejdůležitější periferie ethernetového řadiče a I2C. Tyto budou následně krátce popsány.

3.2.1 Ethernetový modul

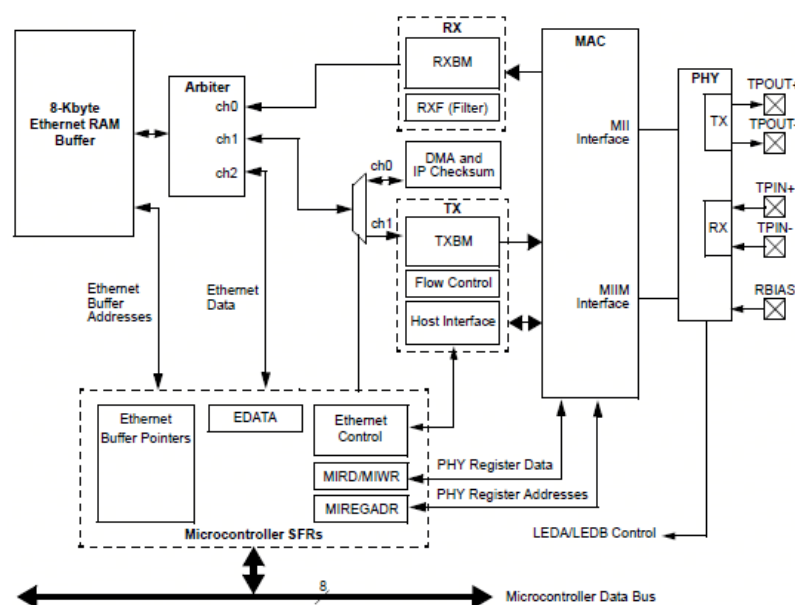
Ethernetový modul implementuje mechanismus komunikace podle standardu IEEE 802.3. Je kompatibilní s 10/100/1000Base-T sítěmi. Podporuje plně duplexní přenos dat rychlostí 10Mbps. Zahrnuje automatické přeposílání paketů při kolizích, automatické odstraňování chybných paketů a generování kontrolního součtu. Spolupracuje s DMA pro rychlé kopírování dat.

Standard IEEE 802.3 specifikuje funkce fyzické a částečně linkové vrstvy (spodní podvrstva) referenčního modelu ISO/OSI. Fyzická vrstva zajišťuje propojení jednotlivých účastníků přenosu a dále obstarává tvorbu elektrického signálu vhodného pro průchod sítí. Linková vrstva implementuje mechanismy nutné pro tvorbu datových rámců, obstarává opakování odesílání při jejich poškození, definuje přístup k médiu metodou CSMA/CD, provádí výpočet kontrolního součtu atd. Tvar rámce je možno vidět na níže uvedeném obrázku.

Preamble 8B	Adresa příjemce 6B	Adresa odesílatele 6B	Délka 2B	DATA 46 - 1500B	Kontrolní součet (CRC) - 4B
-----------------------	------------------------------	---------------------------------	--------------------	---------------------------	---------------------------------------

Obr. 3.4: Rámec podle IEEE 802.3 [3]

Úkolem první části rámce, tedy preamble, je synchronizace vysílací stanice se stanicí přijímací. Další dvě části slouží pro rozlišení adresy příjemce a adresy odesílatele. První 3 byty těchto adres jsou určeny pro identifikaci výrobce zařízení, zbylé slouží pro rozpoznání konkrétního zařízení v rámci výrobce. Pole délka udává délku pole data. (To je oproti protokolu Ethernet II podstatný rozdíl, co se týče definice typu protokolu vyšší vrstvy.) Následující část rámce ukrývá přenášená data, tedy vše od uživatelských dat až po režijní data generovaná protokoly vyšších vrstev. Konečně je připojen kontrolní součet sloužící pro detekci chyb při přenosu.[3]



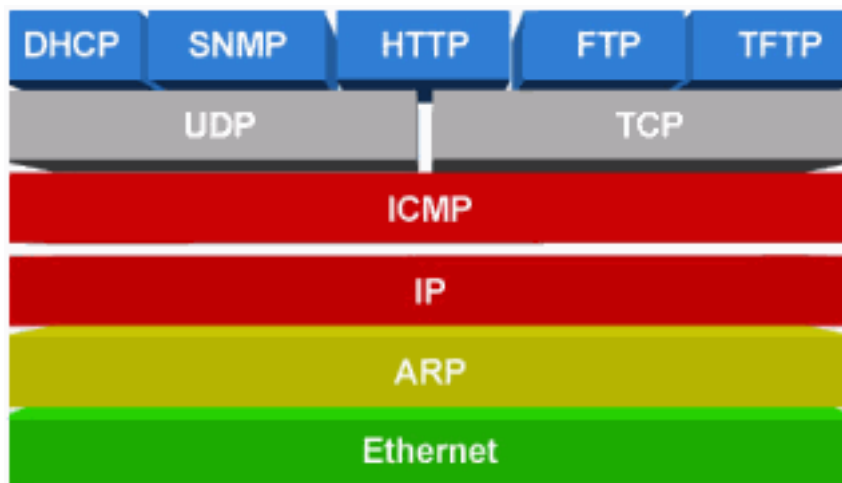
Obr. 3.5: Ethernetový modul mikrokontroléru PIC16F97J60[4]

Na obrázku 3.5 je znázorněno blokové schéma ethernetového modulu. Kromě bloků fyzické (PHY) a linkové vrstvy (MAC) je zde možno vidět 8 Kbytů velkou vyrovnací paměť, která ukládá pakety přijímané a ty co jsou připraveny na odeslání. Je zde také arbitr, který kontroluje přístup do vyrovnávací paměti, pokud je to žádáno od jádra mikrokontroléru nebo DMA. Úkolem sady registrů je řízení předchozích bloků (PHY je řízena prostřednictvím MAC), nastavování ukazatele vyrovnávací paměti a tím pádem zprostředkování rozhraní mezi kódem a hardwarem. Pro detailnější popis je možno nahlédnout do [4].

Microchip TCP/IP stack

Jedná se o sadu protokolů, které realizují funkce vyšších vrstev referenčního komunikačního modelu ISO/OSI, vyjímaje páté a šesté vrstvy. Celkově je kód psán v jazyce C. Pouze pro srovnání s možnostmi na začátku kapitoly je na obrázku 3.6

uveden výčet poskytovaných protokolů. Kromě uvedeného je zde ještě podpora DNS a NBNS. Detailnější popis protokolů by značně navýšil obsah tohoto textu. Případně je možné najít principiální popis jednotlivých vrstev v [3].



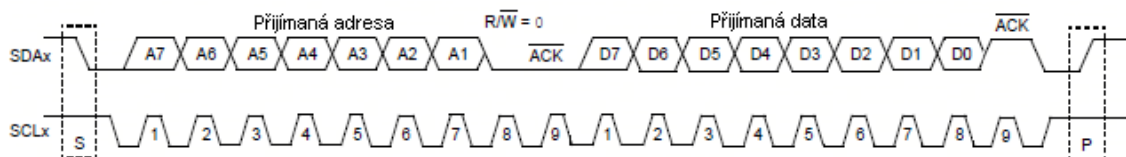
Obr. 3.6: Microchip TCP/IP stack [13]

3.2.2 I2C modul

Periferie I2C je dvou vodičové komunikační rozhraní podporující přenos dat rychlostí 100kbps a 400kbps (případně i 3,4Mbps - použitý mikrokontrolér povoluje pouze předchozí dvě rychlosti [1]). Přenos dat je uskutečněn jedním vodičem (SDA), další vodič přenáší hodinové synchronizační impulsy (SCL). Oba vývody jsou typu otevřený kolektor, proto je třeba připojit zdvihací rezistory. Komunikace probíhá ve vztahu master-slave nebo multimaster, kdy každé zařízení má svoji adresu. Tato adresa může být dlouhá buď 7 nebo 10 bitů.

Přenos dat je vždy započat startovací sekvencí, která je reprezentována specifickou situací na obou vodičích, kdy synchronizační signál je ve stavu logická jedna a na datovém vodiči se vyskytuje sestupná hrana. Ukončovací sekvence probíhá stejným způsobem, pouze s tím rozdílem, že na datovém vodiči je hrana náběžná. Mezi těmito dvěma sekvencemi dochází k přesunu dat po 8 bitových slovech a to nejvíce významným bitem napřed. Tento bitový přenos je opět realizován přechodem logických úrovní, ale v okamžicích, kdy je synchronizační signál ve stavu logická nula. Potřebuje-li master nebo slave čas pro zpracování přijatých dat, podrží po tuto dobu synchronizační kanál ve stavu logické nuly. Část komunikace je znázorněna na obrázku 3.7. Detailnější popis průběhu a časování logických úrovní lze nalézt v [1].

Komunikace na sběrnici je zahájena vždy masterem, kdy po vykonání startovací sekvence následuje 7 bitová adresa, která je zakončena jedním bitem, jež říká zda-li



Obr. 3.7: Část přenosu na I2C [4]

se bude do slave zařízení zapisovat (log. nula) nebo bude-li se z něj číst (log. jedna). Rozpozná-li některé ze zařízení svoji adresu, potvrdí to stáhnutím napěťové úrovně do stavu logické nuly. Následuje přenos dat, než dojde k vyslání ukončovací sekvence.

3.3 Snímač teploty

Vzhledem k tomu, že teplotní snímač DS18B20 komunikuje pomocí jednovodičového rozhraní 1-Wire, bude před popisem snímače toto rozhraní nejdříve charakterizováno.

3.3.1 1-Wire

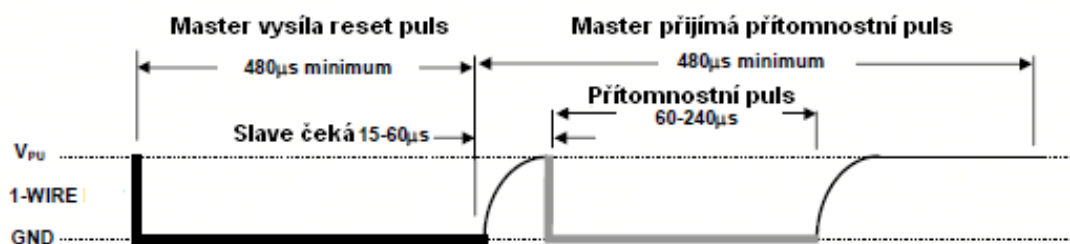
Komunikační rozhraní 1-Wire zajišťuje komunikaci pomocí jednoho datového vodiče. Oproti jiným rozhraním jako je třeba I2C nebo SPI nepotřebuje vodič, po kterém je veden hodinový synchronizační signál. Dále je třeba pro napájení koncových zařízení pouze zemnicí vodič, protože tato zařízení mohou být napájena z datového vodiče. To je zase výhodné, protože obvyklá komunikační rozhraní potřebují navíc vodič s napájecím napětím. Datový vodič je připojen na napájecí napětí 2,8V - 6V přes zvedací rezistor. Rychlost komunikace po této sběrnici může být provozována rychlostí 16,3 kbps a 142 kbps. Jak již bylo řečeno, vzdálenost, na kterou je možno komunikovat se pohybuje okolo stovek metrů. Přesné číslo záleží na počtu připojených zařízení, topologii sítě a na zvolení jedné rychlosti z výše uvedených.

Komunikace s jednotlivými slave zařízeními je realizována pomocí následujících sekvencí signálů.

Resetovací puls a puls oznamující přítomnost snímače na sběrnici

Jakákoliv komunikace s jednotlivými slave zařízeními je zahájena masterem, který vyšle na sběrnici resetovací puls. Po uplynutí doby minimálně $480\mu s$ master přestane resetovací puls vysílat a přejde do příjímacího módu. Snaha zdvihacího rezistoru dostat napěťovou úroveň signálu na datovém vodiči do stavu logické jedničky vytvoří náběžnou hranu. Detekuje-li slave tuto hranu, počká $15\mu s$ - $60\mu s$ než odešle puls

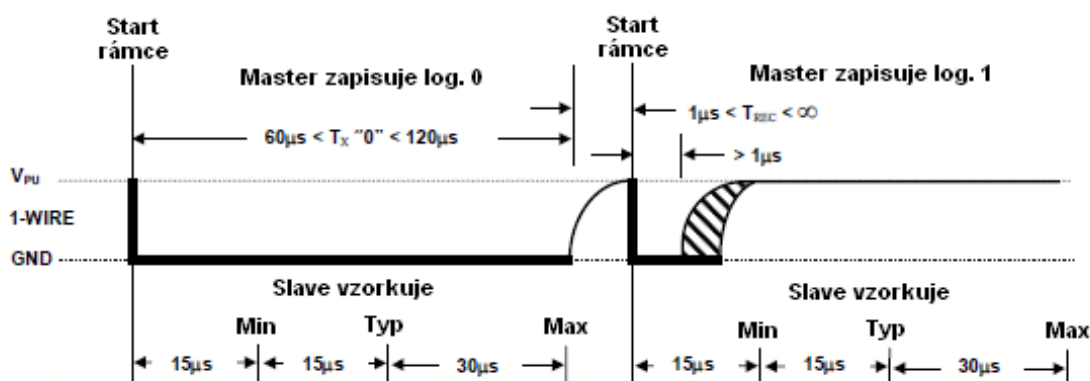
oznamující přítomnost snímače na sběrnici, který je reprezentován podržením sběrnice ve stavu logické nuly po dobu $60\mu s - 240\mu s$.



Obr. 3.8: Resetovací puls [5]

Zápis jednoho bitu

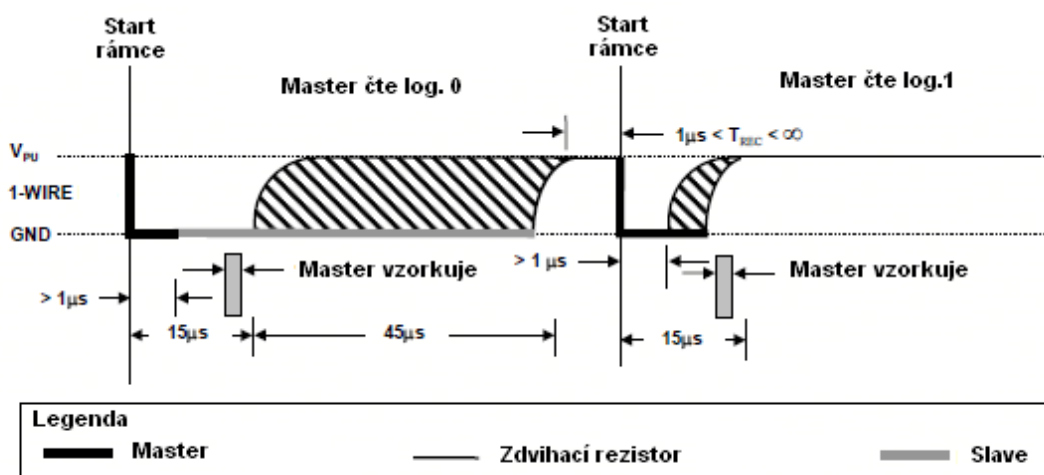
Provede-li se úspěšně předchozí sekvence pulsů, můžeme následně začít vysílat požadovaná data. To se děje pomocí časových rámců o celkové době trvání $60\mu s - 120\mu s$. Podrží-li master napěťovou úroveň na nule právě po dobu trvání tohoto rámce, dojde tak k zápisu logické nuly na sběrnici. Slave zařízení tuto hodnotu detekuje vzorkováním napěťové úrovně na sběrnici nejdříve $15\mu s$ po startu časového rámce. Zápis logické jedna probíhá tak, že master vyšle puls, který dostane napěťovou úroveň na sběrnici na nulu a po dobu nejméně $1\mu s$ ji tam udrží, následně zdvihací rezistor vrátí napětí zpět nahoru a v tomto stavu sběrnice setrvá až do konce časového rámce. Slave zařízení opět vzorkuje napěťovou úroveň nejméně po $15\mu s$ od začátku časového rámce, čímž přečte logickou jedničku.



Obr. 3.9: Zápis jednoho bitu [5]

Čtení jednoho bitu

Čtení jednotlivých bitů je prováděno stejně jako u zápisu pomocí časových rámců, které trvají $60\mu\text{s}$ - $120\mu\text{s}$. Logická hodnota signálu na sběrnici je přečtena tak, že master nejprve přidrží napěťovou úroveň signálu na nule po dobu nejméně $1\mu\text{s}$. Následně tuto hodnotu uvolní a podle toho, zda-li slave zařízení napětí drží stále na nule nebo zdvihací rezistor napětí dostane na horní úroveň, může master na sběrnici navzorkovat logickou nulu nebo jedničku. Vzorkování master provádí po uplynutí $15\mu\text{s}$ od začátku rámce nebo o něco dříve, jak je uvedeno na obrázku 3.10.



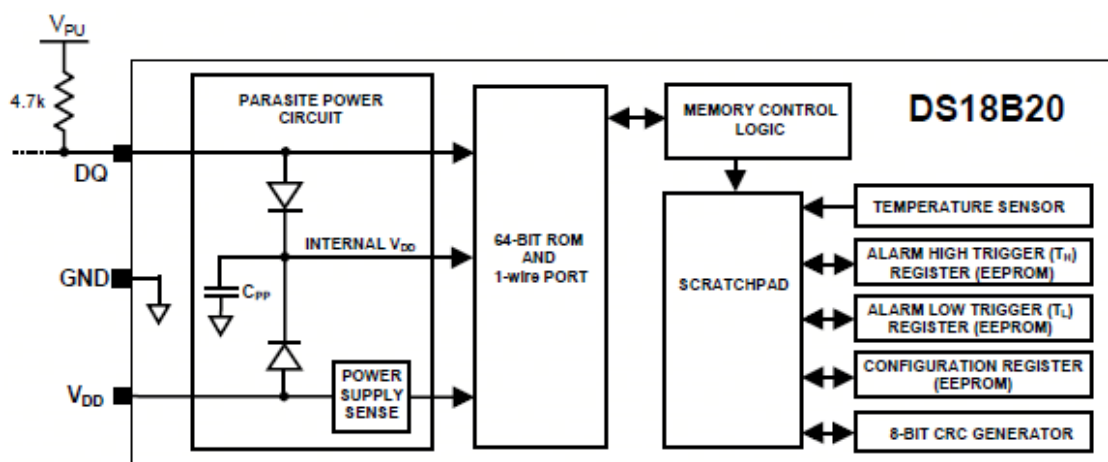
Obr. 3.10: Čtení jednoho bitu [5]

Maximální počet připojitelných zařízení

Rozhraní 1-Wire jednoznačně nedefinuje maximální počet připojitelných zařízení. Vzhledem k tomu, že každé zařízení obsahuje unikátní 48-bitový identifikační kód (bude uveden dále), je maximální počet připojitelných zařízení dán spíše parametry vedení a velikostí proudu, který jsme schopni na rozhraní dodat. Velikostí proudu z toho důvodu, že některá zařízení vyžadují pro své operace proud například $1,5\text{mA}$. Pokud chceme u dále uvedeného snímače teploty měřit současně u všech připojených snímačů, je jasné, že bude v tento okamžik třeba na rozhraní dodávat proud tolikrát větší, než je počet připojených zařízení. Na druhou stranu i když zajistíme dostatečnou proudovou podporu, tak jsou zde stále věci jako délka vedení, jeho kapacita apod.

3.3.2 DS18B20

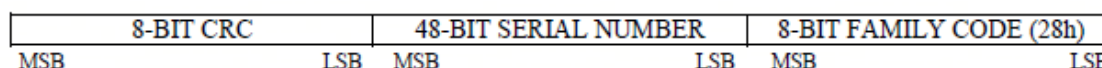
DS18B20 je digitální teplotní snímač vyráběný firmou Maxim, který komunikuje po jednovodičové sběrnici Dallas. Podporovaný rozsah měření teploty je -75°C až 125°C . Měření s přesností 0.5°C probíhá v rozsahu -10°C až 85°C . Každý teplotní snímač je označen unikátním 64-bitovým ROM kódem, což umožňuje jeho identifikaci. Převod teplota/číslo může být prováděno s rozlišením 9, 10, 11 a 12 bitů, přičemž nejdelší doba převodu pro 12-bitové rozlišení je 750ms. Snímač je vyráběn ve třech variantách pouzder jako 8-pinové SO, μSOP a 3-pinové TO-92.



Obr. 3.11: Blokové schéma snímače teploty DS18B20 [5]

Na blokovém schématu uvedeném na obrázku 3.11 si lze všimnout obvodu parazitního napájení, který slouží k výběru jednoho ze dvou módů napájení. Prvním módem je myšlen normální způsob napájení, kdy na pin V_{DD} je přivedeno napětí 3V až 5.5V. Druhým módem je parazitní způsob napájení. V tomto módu je snímač napájen z datového vodiče sběrnice, nachází-li se ve stavu H. Pokud klesne napětí na úroveň L přebírá roli napájecího zdroje interní kondenzátor C_{PP} . Používáme-li parazitní způsob napájení je třeba připojit pin V_{DD} na zem.

Jako další je na obrázku uveden blok zahrnující hardwareovou implementaci 1-Wire rozhraní a 64-bitového ROM kódu. Struktura tohoto kódu je na obrázku 3.12.



Obr. 3.12: ROM kód DS18B20 [5]

Osm nejvíce významných bitů slouží pro uchování kontrolního součtu, který je vypočten ze zbylých 56 bitů kódu. Nejméně významných osm bitů slouží pro rozčlenění typů zařízení do rodin, aby bylo možno identifikovat, zda-li se jedná například o paměť, termostat a další skupiny zařízení komunikujících po 1-Wire, které firma Maxim poskytuje.

Soubor registrů (Scratchpad), tak jak je uveden na obrázku 3.13, se skládá z 8 bytů. První dva byty představují prostor pro ukládání měřené teploty. Z této dvojice bytů vyjadřuje prvních jedenáct méně významných bitů teploty ve dvojkovém doplňku. Zbylých pět bitů představuje znaménko. Třetí a čtvrtý byte slouží pro uložení dolní a horní meze teploty. Tyto teploty jsou porovnávány s aktuálně uloženou teplotou v prvních dvou registrech (při každém měření). Jestliže jsou stejné je možno použitím příslušné instrukce přechist stav překročení těchto mezí. Pátý byte slouží jako konfigurační registr snímače. Následující trojice bytů je rezervována pro interní užití samotným snímačem. Poslední byte reprezentuje kontrolní součet potřebný pro ohlídání bezchybného přenosu celého souboru výše uvedených registrů, používáme-li instrukci čtení všech těchto registrů najednou.

SCRATCHPAD (Power-up State)	
byte 0	Temperature LSB (50h) } (85°C)
byte 1	Temperature MSB (05h) }
byte 2	T _H Register or User Byte 1*
byte 3	T _L Register or User Byte 2*
byte 4	Configuration Register*
byte 5	Reserved (FFh)
byte 6	Reserved (0Ch)
byte 7	Reserved (10h)
byte 8	CRC*

Obr. 3.13: Soubor registrů DS18B20 [5]

Příkazy pro komunikaci se snímačem

Veškeré příkazy využívající funkce snímače se dělí na dvě kategorie. V následujícím je za názvem příkazu v hranatých závorkách uveden kód příkazu.

První kategorie obsahuje příkazy potřebné pro práci s pamětí ROM.

- Vyhledej ROM [F0h] - příkaz, který slouží pro identifikaci všech zařízení připojených na sběrnici.
- Čti ROM [33h] - obstarává čtení ROM kódu pouze jednoho zařízení připojeného na sběrnici.

- Porovnej ROM [55h] - vyšleme-li tento příkaz na sběrnici, dojde k postupnému odpojování jednotlivých zařízení až zůstane pouze poslední vybrané. Do tohoto můžeme následně vysílat příkazy pro práci s registry.
- Přeskoč ROM [CCh] - nedojde k proceduře porovnávání ROM jako v předchozím případě. Tohoto příkazu je využito tehdy, chceme-li použít některý z příkazů pracující s registry u všech zařízení najednou. Dále je používán, je-li na sběrnici pouze jedno zařízení.
- Vyhledej poplach [ECh] - tento příkaz plní stejnou funkci jako vyhledávání ROM pouze s tím rozdílem, že odpovídají jen ta zařízení, u kterých došlo k překročení mezních teplot nastavených v registrech k tomu určených (viz. soubor registrů).

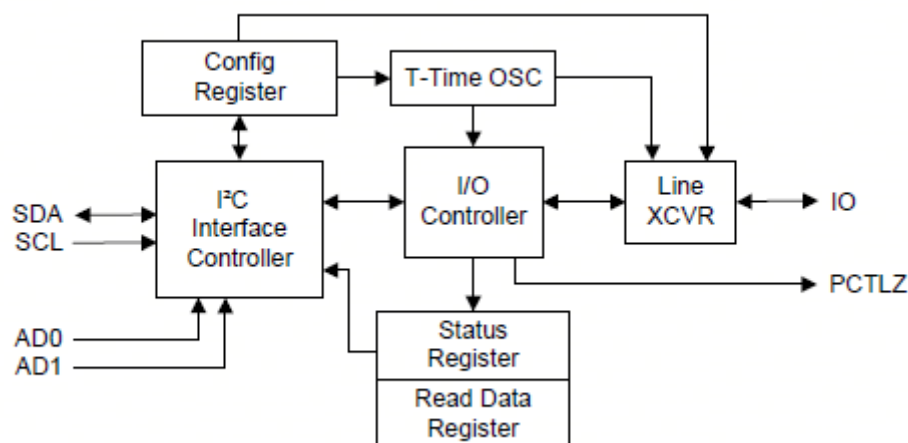
Druhá kategorie obsahuje příkazy pro práci se skupinou registrů.

- Převeď teplotu [44h] - tento příkaz provede jedno měření teploty. Výsledek bude uložen do prvních dvou registrů snímače, jak bylo uvedeno v předchozím. Je-li používáno parazitní napájení je třeba dodávat po dobu vykonávání tohoto příkazu dostatek proudu na sběrnici. Je-li převod ukončen čeká zařízení na RESET puls.
- Zapiš obsah registrů [4Eh] - tato instrukce provádí zápis do tří registrů. Jedná se o T_H registr, T_L registr a konfigurační registr. Data jsou odesílána nejméně důležitým bitem napřed.
- Čti obsah registrů [BEh] - čte obsah registrů od prvního obsahující měřenou teplotu až po poslední obsahující kontrolní součet. Není nutné čekat pokaždé na přečtení všech registrů, kdykoliv je možno čtení ukončit RESET pulsem.
- Kopíruj obsah registrů [48h] - provede překopírování obsahu registrů T_L , T_H a konfiguračního do E2PROM. Při provádění této instrukce je třeba proudově posílit sběrnici na dobu 10ms, používáme-li parazitního napájení.
- Obnov E2PROM [B8h] - opačná operace oproti předchozí instrukci. Dojde k překopírování obsahu E2PROM do souboru registrů. Po dobu vykonávání tohoto příkazu se sběrnice drží na log. 0. Vykonávání je dokončeno přejde-li sběrnice do log. 1.
- Zjistí způsob napájení [B4h] - instrukce zjistí zda-li je příslušné zařízení na sběrnici napájeno normálním nebo parazitním způsobem napájení. Objeví-li se na sběrnici po vyslání tohoto příkazu nízká úroveň napětí, jedná se o parazitní způsob napájení, jestliže zůstane napěťová úroveň na sběrnici na vysoké úrovni, jedná se o normální způsob napájení.

3.3.3 DS2482-100

Vzhledem tomu, že vybraný mikrokontrolér neobsahuje žádné rozhraní pro komunikaci po 1-Wire a softwareová realizace instrukcí uvedených v předchozí podkapitole je časově náročnější na odladění, bylo rozhodnuto přiřadit do výsledné realizace obvod DS2482-100, který již obsahuje interface pro 1-Wire.

DS2482-100 je obvod, který v rolích master/slave na sběrnici 1-Wire obstarává roli mastera. Jedná se o převodník z rozhraní I2C na 1-Wire. Rychlost přenosu po I2C je podporována ve standardní 100kbps a v rychlejší 400kbps verzi. Rychlost komunikace po 1-Wire je možno uskutečnit ve verzi 16.3kbps a 142kbps. Napájení je možné v rozsahu 2.9V až 5V.



Obr. 3.14: Blokové schéma DS2482-100 [6]

Hlavním obvodem, který je uvedený v blokovém schématu na obrázku 3.14 je I/O kontrolér. Právě tento obvod obstarává generování časově kritických funkcí komunikačního protokolu 1-Wire. Dále se zde nachází obvod pro I2C komunikaci s nadřazeným mikrokontrolérem, interní oscilátor a tři registry.

První z registrů je konfigurační, jeho znázornění je uvedeno na obrázku 3.15. Po startu zařízení je v registru zapsána hodnota 00h. Chceme-li do tohoto registru

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
1WS	SPU	PPM	APU	1WS	SPU	PPM	APU

Obr. 3.15: Konfigurační registr DS2482-100 [6]

zapsat nová data, lze tak učinit pouze tehdy jsou-li horní čtyři bity jednotkovým

doplňkem spodních čtyř. Sedmý bit 1WS slouží k volbě rychlosti komunikace po 1-Wire. Zápisem logické nuly do tohoto bitu zvolíme rychlost 16.3kbps, naproti tomu zápisem logické jedničky zvolíme 142kbps. Šestý bit při logické jedničce zapne interní pullup tranzistor, čímž dojde k ovlivnění strmosti hran signálu na 1-Wire sběrnici a zároveň k proudovému posílení sběrnice. Pátý bit PPM zkracuje dobu sestupné hrany pulsu oznamujícího přítomnost snímače na sběrnici (funguje pouze pro menší z výše uvedených rychlostí). Jestliže nebude vnitřní pullup tranzistor stačit na posílení sběrnice, lze využít pin PCTLZ pro kontrolu externího MOSFET tranzistru. To se děje pomocí čtvrtého bitu SPU.

Jako druhý je registr stavový, který je určen pouze pro čtení. Jeho zobrazení je na následujícím obrázku 3.16. Nejméně významný bit 1WB plní funkci indikace zaneprázdněnosti sběrnice. Další bit PPD se nachází ve stavu logická jedna, je-li detekován puls oznamující přítomnost snímače na sběrnici. Bit SD detekuje stav horní úrovně na sběrnici před pulsem oznamujícím přítomnost snímače na sběrnici. LL podává zprávu o stavu logické úrovně na sběrnici bez provedení inicializace sběrnice. RST signalizuje, že zařízení bylo restartováno, a to do doby než dojde k zápisu do konfiguračního registru. SBR slouží pro detekci logické úrovně na 1-Wire, po použití příkazu 1-Wire bit a 1-Wire triplet. TSB podává zprávu o stavu logické úrovně na 1-Wire sběrnici ve druhé periodě příkazu 1-Wire triplet. Poslední DIR bit je využíván v souvislosti s vyhledávacím algoritmem ROM kódů. Detailnější popis je uveden v [6].

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
DIR	TSB	SBR	RST	LL	SD	PPD	1WB

Obr. 3.16: Stavový registr DS2482-100 [6]

Příkazy pro komunikaci s DS2482-100

DS2482-100 komunikuje pomocí 8 příkazů, které se dělí do čtyř kategorií. Jedná se o příkazy pro ovládání zařízení, I2C komunikaci, nastavení 1-Wire rozhraní a 1-Wire komunikaci. V následujícím je za názvem příkazu v hranatých závorkách uveden kód příkazu.

- Restart zařízení [F0h] - restartuje DS2482-100. Při vyslání tohoto příkazu dojde k přerušení právě probíhající komunikace na 1-Wire. Instrukce nemá žádný vstupní parametr.

- Nastavení ukazatele čtení [E1h] - nastaví ukazatele čtení na vybraný registr. Vstupní parametrem této instrukce je hodnota F0h pro stavový registr, E1h pro datový registr a C3h pro konfigurační registr.
- Zápis konfigurace zařízení [D2h] - Zapiše novou konfiguraci zařízení do konfiguračního registru. Změna konfigurace se projeví okamžitě po zapsání. Vstupním parametrem je zde konfigurační slovo, které bude přijato pouze tehdy, jsou-li první čtyři nejvýznamnější bity jednotkovým doplňkem čtyř nejméně významných bitů.
- 1-Wire restart [B4h] - instrukce vyvolá restart na sběrnici 1-Wire. Tato instrukce nemá vstupní parametr.
- 1-Wire bit [87h] - Vypíše na sběrnici jeden bit. Vstupním parametrem je jeden byte z něhož ten nejvýznamnější bit určuje hodnotu zapsaného bitu na 1-Wire sběrnici.
- Zápis jednoho bytu [A5h] - zapiše jeden byte na 1-Wire sběrnici. Vstupním parametrem je zapisovaný byte.
- Čtení jednoho bytu [96h] - přečte jeden byte z 1-Wire sběrnice. Tato instrukce nemá vstupní parametr.
- 1-Wire triplet [78h] - instrukce vygeneruje tři časové intervaly na 1-Wire sběrnici o stejné délce jako při použití tří 1-Wire bitů, kde dva budou pro čtení a jeden pro zápis. Této instrukce je využíváno při vyhledávání ROM kódů.

4 NÁVRH ZAŘÍZENÍ

Tato kapitola je rozdělena na dvě části. První se zabývá schematickým návrhem zařízení a ta druhá návrhem plošného spoje. Pro návrh byl využit návrhový systém Eagle.

4.1 Obvodový návrh

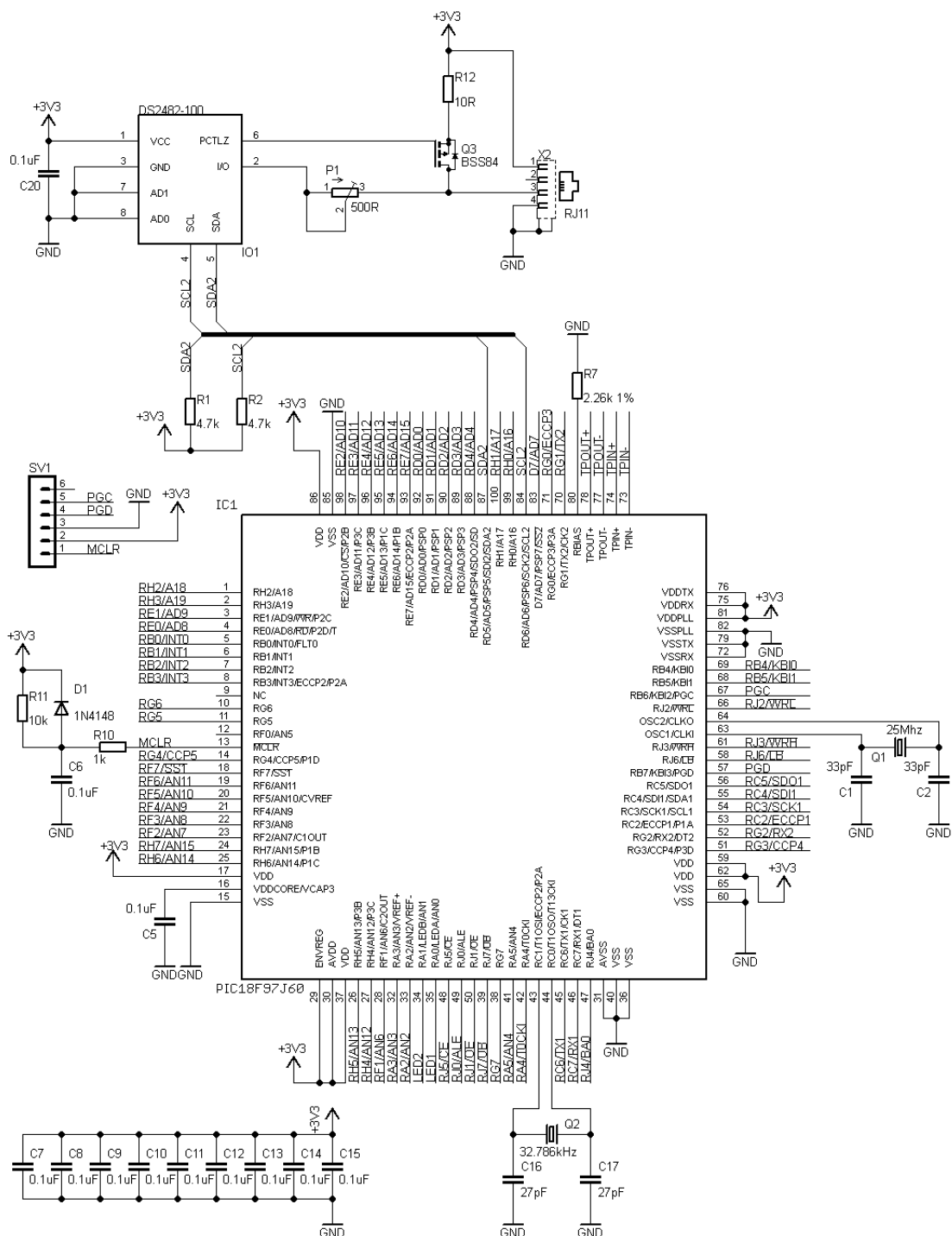
V následujícím textu jsou popsány jednotlivé části zapojení. Celkové schéma zapojení je uvedeno v příloze na konci tohoto dokumentu.

4.1.1 Zapojení mikrokontroléru

Zapojení mikrokontroléru je uvedeno na obrázku 4.1. V levém dolním rohu je možno vidět řadu kondenzátorů, která je připojena mezi kladné napájecí napětí a zem. Tyto kondenzátory je třeba při návrhu plošného spoje umístit na každý pár napájecích svorek mikrokontroléru. Funkce těchto kondenzátorů spočívá v přidržování stálé úrovně napájecího napětí na mikrokontroléru, která by mohla být ovlivňována impulsními proudy vznikajícími vlivem impulsního odběru okolních zařízení. K tomu by mohlo přispět také synchronní překlápění většího počtu hradel v mikrokontroléru a to zejména kdyby byl využit port B nebo C, které poskytují větší hodnoty proudu pro odběr. Hodnota těchto kondenzátorů je zvolena dle výrobce na 100nF. Kondenzátor C5 připojený na pin VDDCORE/VCAP povoluje funkci vnitřního stabilizátoru napětí v mikrokontroléru.

Vzhledem k tomu, že při startu mikrokontroléru trvá nějakou dobu než napájecí napětí dorazí na všechny napájecí svorky mikrokontroléru a nabije kondenzátory na nich umístěné, je na resetovací MCLR (Master Clear Reset) pin připojen obvod složený z rezistorů R10, R11 a kondenzátoru C6. Úkolem tohoto obvodu je nastartování mikrokontroléru až v té době, kdy je napájecí napětí všude kde má být. Nepřidání tohoto zapojení by mělo neblahý vliv na funkci mikrokontroléru při jeho startu, což by se mohlo projevit i na jeho následné funkci. Dioda D1 obstarává rychlé vybití kondenzátoru C6. To je důležité proto, aby mikrokontrolér nefungoval ještě v době kdy dochází k vybíjení postranních kondenzátorů a kondenzátorů připojených na výstupu stabilizátoru napětí.

V zapojení je dále dvojice oscilátorů Q1 a Q2, které pracují v Colpittsově zapojení. Oscilátor Q1 zde plní funkci primárního oscilátoru pracujícího na kmitočtu 25MHz. Tato hodnota musí být dodržena z důvodu využívání ethernetového řadiče mikrokontroléru. Oscilátor Q2 je sekundární oscilátor připojený na obvod prvního



Obr. 4.1: Zapojení mikrokontroléru a převodníku DS2482-100

časovače mikrokontroléru (Timer1), kterého se dá využít chceme-li mít ve své aplikaci hodiny reálného času. Kmitočet 32,786 kHz není náhodný. Je využit proto, aby bylo možno dosáhnout po patnácti dělení dvěma kmitočtu 1Hz.

Rezistor R7 je připojen v souvislosti s ethernetovým modulem. Jeho přítomnost

má vliv na amplitudy signálů u pinů TPOUT+ a TPOUT-. Dle výrobce je zvolena hodnota 2.26k s tolerancí 1%.

Na každý vývod druhé I2C sběrnice mikrokontroléru SCL2 a SDA2 je připojen jeden pull-up rezistor. Hodnota těchto rezistorů byla zvolena na 4.7k.

Konečně je na piny PGC, PGD a MCLR připojen vývod pro programátor.

4.1.2 Připojení snímačů DS18S20

Připojení snímačů je provedeno pomocí obvodu DS2482-100, který (jak již bylo řečeno) komunikuje s nadřazeným mikrokontrolérem pomocí I2C rozhraní. Schema zapojení této části je uvedeno spolu s mikrokontrolérem na obrázku 4.1. Tato část schematu byla převzata z dokumentace k obvodu DS2482-100. 1-Wire rozhraní je z tohoto obvodu vyvedeno na konektor RJ11, ke kterému se připojí snímače teploty. Dále je na pin PCTLZ připojen unipolární MOSFET tranzistor zajišťující proudové posílení datového vodiče. To je nutné jestliže používáme u senzorů parazitní způsob napájení a chceme jich připojit více, protože při některých operacích odebírá každé proud až 1.5mA. Výstupní pin 1-Wire rozhraní poskytuje maximálně 20mA, což by mohlo být omezující. Pomocí MOSFET tranzistoru tedy navýšíme možný počet připojených zařízení. Na napájecí svorky obvodu DS2482-100 je připojen lokální blokovací kondenzátor, zajišťující přidržování napěťové úrovně při impulsním odběru proudu. Potenciometr P1 plní funkci zakončovacího rezistoru, který redukuje odrazy na vedení. Jeho hodnota není definitivní a bude doladěna až při realizaci zařízení. Bude se jednat o hodnotu přibližně 100Ω jak výrobce doporučuje. Rezistor R12 určuje maximální proud dodávaný na sběrnici. Prahové napětí tranzistoru Q3 je dle jeho výrobce maximálně 2V. Hodnotu rezistoru R12 je tedy třeba navrhnout tak, aby bylo toto napětí dodrženo. Napětí U_{GS} bude tedy zvoleno na hodnotu 2,7V čímž je prahové napětí dostatečně překročeno. Dle výrobce tranzistoru odpovídá tomuto napětí U_{GS} hodnota proudu přibližně 60mA procházející tranzistorem. Napětí 2,7V se potom rozloží mezi elektrody GS a zbytek do 3,3V zůstane na rezistoru. (Důležité je připomenout, že tranzistor je zapojen elektrodou source na rezistor R12.) Potom bude výsledná hodnota rezistoru R12:

$$R_{12} = \frac{U_N - U_{GS}}{I_D} = \frac{3,3 - 2,7}{0,06} = 10\Omega \quad (4.1)$$

U_{GS} - napětí mezi elektrodami GS

U_N - napájecí napětí

I_D - proud tekoucí tranzistorem

4.1.3 Ethernetový transformátor

Hlavním účelem ethernetového transformátoru je galvanické oddělení ethernetového modulu mikrokontroléru od sítě ethernet. V zapojení se nachází feritová perla, která slouží pro omezení elektromagnetického vyzařování. Rezistory R3 až R6 zde plní funkci zakončovací rezistorů. Zapojení je převzato od výrobce mikrokontroléru. Napětí na diodách je dle údajů výrobce transformátoru 2,5V pro zelenou a 2,1V pro žlutou. S uvážením těchto hodnot a volbou proudu 10mA každou diodou byly navrhnuty hodnoty rezistorů podle níže uvedeného výpočtu. Hodnota 80Ω je zaokrouhlena na 82Ω z důvodu dostupnosti rezistoru této hodnoty.

$$R_8 = \frac{U_N - U_{DZE}}{I_D} = \frac{3,3 - 2,5}{0,01} = 80\Omega \quad (4.2)$$

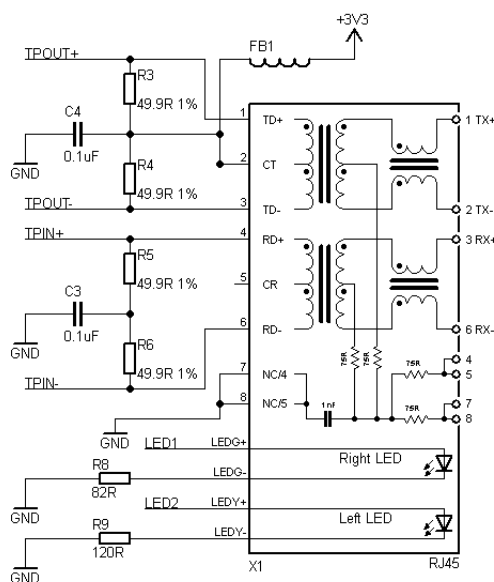
U_{DZE} - napětí na zelené diodě

U_N - napájecí napětí

I_D - proud tekoucí diodou

$$R_9 = \frac{U_N - U_{DZL}}{I_D} = \frac{3,3 - 2,1}{0,01} = 120\Omega \quad (4.3)$$

U_{DZL} - napětí na žluté diodě



Obr. 4.2: Zapojení ethernetového transformátoru

4.1.4 Napájení

Veškeré komponenty zapojení potřebují pro svoji funkci napájecího napětí 3,3V, což je prvním kritériem pro volbu vhodného stabilizátoru napájecího napětí. Dalším kritériem je celkový odběr těchto komponent. Ten je vypočten podle níže uvedeného vztahu:

$$I_{MAX} = I_M + I_{Meth} + I_{MPA0} + I_{MPA1} + I_{DS2482} + I_{T1} = 21,6 + 214 + 10 + 10 + 20 + 60 = 335,6mA \quad (4.4)$$

I_M - maximální odběr mikrokontroléru při vnitřním taktu 41.6667MHz

I_{Meth} - maximální odběr ethernetového modulu mikrokontroléru

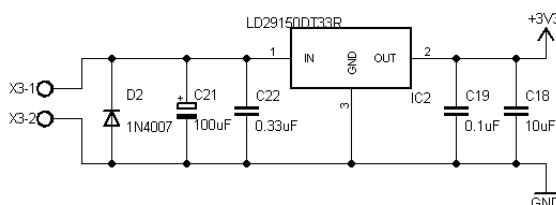
I_{MPA0} - odběr zelené LED diody ethernetového transformátoru

I_{MPA1} - odběr červené LED diody ethernetového transformátoru

I_{DS2482} - maximální odběr DS2482-100

I_{T1} - odběr tranzistoru T1 v saturaci

Stabilizátorem napětí, který pokryje výše uvedené potřeby více než dostatečně, je obvod LD29150#33 od firmy STMicroelectronics. Jedná se o stabilizátor s výstupním napětím 3.3V, maximálním zatížením 1.5A a nízkým úbytkem napětí 0.4V.



Obr. 4.3: Zapojení napájecí části

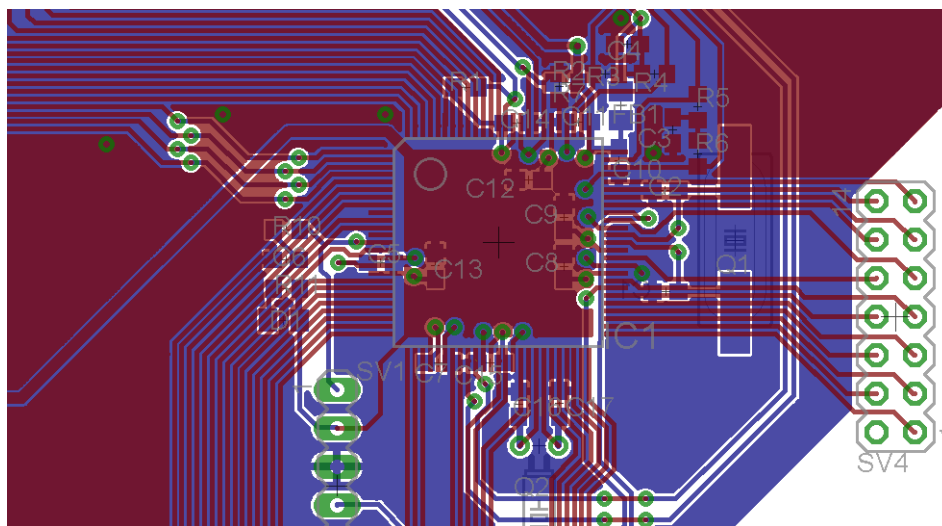
V okolí stabilizátoru si lze všimnout několika kondenzátorů. Vezmeme-li to zleva, tak C21 je zde z důvodu potlačení parazitní indukčnosti a odporu přívodů od transformátoru napětí, který bude veden ke stabilizátoru delším vodičem. C21 je volen jako elektrolytický kondenzátor s hodnotou pohybující se okolo stovek uF. Samotný kondenzátor C21 by neměl příliš dobré vysokofrekvenční vlastnosti a proto je doplněn keramickým kondenzátorem C22. Hodnota C22 se volí obvykle v rozsahu 100-330nF. Na doporučení výrobce byla zvolena hodnota 330nF. Kondenzátor C19 je důležitý vzhledem k jeho vlivu na amplitudu a kmitočet při impulsním zatížení obvodu. C18 plní funkci globálního filtračního kondenzátoru, který drží úroveň napětí na výstupu stabilizátoru vždy v okamžiku impulsního odběru proudu. Jedná se o tantalový kondenzátor s hodnotou 10uF. Tantalový kondenzátor je zvolen z důvodů jeho nízké parazitní indukčnosti. Na vstupní straně stabilizátoru je dále zapojena dioda v závěrném směru, která chrání stabilizátor proti přepólování napájecího napětí.[12]

4.2 Návrh plošného spoje

Každý návrh plošného spoje by měl začít rozvahou o rozmístění součástek. Při tom je důležité, aby jednotlivé komponenty zapojení byly rozmístěny po povrchu desky se zřetelem na jejich funkci. Je tedy třeba rozlišovat podle toho, zda-li se jedná o součástky výkonové, analogové, číslicové a nebo ty, které jsou jejich kombinací. Podle toho se potom snažíme desku plošného spoje rozdělit na sektory, ve který se nachází shluky těchto součástek. Dále je třeba brát ohled na frekvence, se kterými dané zařízení pracuje a také na mechanické uspořádání konektorů a ovládacích prvků.

4.2.1 Číslicová část

Povšimneme-li si části realizace plošného spoje na obrázku 4.4, je zde možno vidět několik prokovení pod mikrokontrolérem. Pomocí těchto prokovení jsou připojeny z druhé strany desky dříve diskutované kondenzátory pro blokování napájení na napájecích pinech mikrokontroléru. Právě toto provedení je nutností s ohledem k tomu, že kdyby se provedlo jejich připojení ze strany mikrokontroléru, bylo by třeba je od něj umístit ve větší vzdálenosti. Právě s rostoucí vzdáleností spojů by se vytvářela zbytečně velká proudová smyčka a při impulsním odběru proudu z těchto blokovacích kondenzátorů by docházelo k vyzařování elektromagnetického pole.



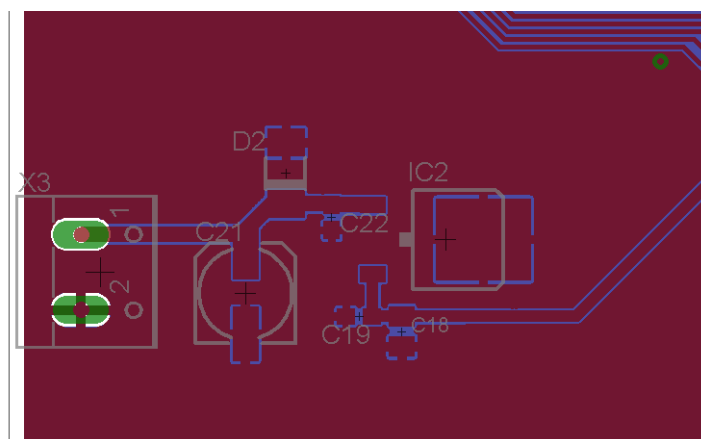
Obr. 4.4: Deska plošného spoje v oblasti procesoru

Přímo pod mikrokontrolérem lze vidět celistvou plochu kladného napájecího napětí, která spolu s rozlévanou zemí z druhé strany opět pod mikrokontrolérem tvoří příznivou kapacitu. Dále je tato plocha schopna pojmout malé množství ztrátového tepla mikrokontroléru.

Obvod zdroje hodinového signálu Q1 je pomocí dvou prokovených otvorů veden na druhou stranu desky, aby se docílilo snížení vlivu přeslechů na vstupně výstupní obvody. Pro vytvoření kvalitního taktovacího obvodu je také dobré dodržet stejnou délku obou přívodů a také jejich konstantní šířku. Také je vhodné kolem krystalového rezonátoru rozlít zem, čímž dojde opět k potlačení vyzařování.

4.2.2 Napájecí část

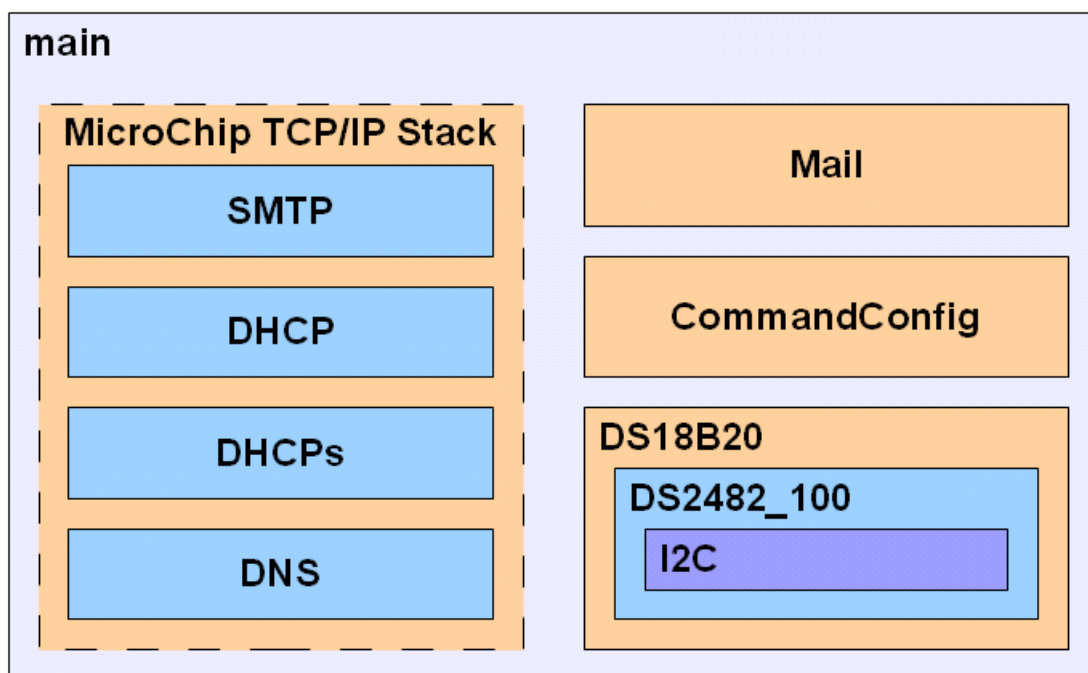
Napájecí část desky plošného spoje je místo, kde se sbíhají všechny proudy od všech zařízení na celé desce plošného spoje. Proto je třeba aby tato část byla oddělena od všech ostatních. Nedodržením této zásady by mohlo způsobit ovlivnění okolních částí obvodu elektromagnetickým vyzařováním obvodu stabilizátoru. Veškeré součástky na obrázku 4.5 jsou voleny jako SMD z důvodu snížení parazitní indukčnosti přívodů. Kondenzátory v okolí stabilizátoru je třeba umístit pokud možno co nejblíže, aby byly sníženy plochy proudových smyček. To je důležité z důvodu opětovného snížení elektromagnetického vyzařování do okolí a také pro snížení parazitní indukčnosti, která by mohla mít vliv na velikost zákmitů na výstupu stabilizátoru při impulsním odběru proudu z těchto kondenzátorů. Okolní vodivá plocha slouží pro pojmутí ztrátového tepla stabilizátoru. Protože z výstupu stabilizátoru tečou větší proudy, než v ostatních částech zařízení, je třeba vodič na tomto výstupu rozšířit.



Obr. 4.5: Deska plošného spoje v oblasti napájení

5 SOFTWAREVÉ VYBAVENÍ

Tato kapitola se bude věnovat popisu softwarového vybavení mikrokontroléru. V úvodu bude naznačena celková hierarchie firmware mikrokontroléru. Dále budou popsány rutiny používané pro komunikaci se snímači, bude objasněno odesílání emailu a konečně také ovládání a konfigurace přístroje.



Obr. 5.1: Hierarchie modulů softwarového vybavení mikrokontroléru

Na obrázku 5.1 je uvedena hierarchie modulů softwarového vybavení mikrokontroléru. Levá část obrázku znázorňuje moduly, které jsou využívány v Microchip TCP/IP stacku. Jedná se pouze o moduly aplikační vrstvy. Pro jejich správnou funkci je třeba ještě mnoha dalších modulů nižších vrstev, které zde nejsou uvedeny z důvodu přehlednosti obrázku. Jako první je uveden SMTP modul zajišťující odesílání emailu. Druhý je zde DHCP klient sloužící pro získání IP adresy z nadřazeného systému (DHCP serveru jiného zařízení). Předposlední je DHCP server sloužící pro přidělování IP adres zařízením, která obsahují DHCP klient. Poslední modul obstarává funkce DNS klienta, který zajišťuje transformaci adresy ve formě řetězce na IP adresu.

V pravém horním rohu je možno vidět modul zajišťující periodické měření teploty, sestavování emailové zprávy a její odesílání.

Jako další je zde modul pro konfiguraci a ovládání přístroje a snímačů teploty. Toto ovládání a konfigurace se děje za pomoci dvou skupin příkazů, které budou

uvedeny později.

Soubor posledních tří modulů implementuje komunikaci mezi mikrokontrolérem a snímači teploty. Každý z modulů nese jméno zařízení nebo periferie, kterou obsluhuje. Na nejnižší vrstvě je modul I2C, který obsluhuje stejnojmennou periferii mikrokontroléru a obstarává tak komunikaci s fyzickým zařízením DS2482-100. O úroveň výše je modul DS2482_100, který obsahuje rutiny potřebné pro ovládání tohoto zařízení a pro komunikaci po 1-Wire rozhraní. Jako poslední je na nejvyšší vrstvě modul DS18B20, který obsahuje softwarovou reprezentaci snímačů teploty a sadu obslužných funkcí.

5.1 Hlavní program

Vytvoření hlavního programu s Microchip TCP/IP stackem je výhodné udělat jako první věc, protože je tak možné využít TCP spojení a pomocí serverové aplikace běžící v mikrokontroléru odladit rutiny pro komunikaci se snímači teploty. Na tomto místě bude popsáno, jak nakonfigurovat a vytvořit aplikaci využívající Microchip TCP/IP stack. Bude zde v krátkosti objasněn princip funkce tohoto stacku a konečně bude znázorněn a popsán hlavní program mikrokontroléru.

5.1.1 Konfigurace stacku

Aby bylo možné provozovat ve vlastní aplikaci Microchip TCP/IP stack, je třeba udělat několik věcí, mezi které patří i konfigurace. To se provádí za pomoci modulu TCPIPConfig.h, kde hned na začátku tohoto modulu je je možno za pomoci komentáře umožnit vložení (include) těch modulů, které je třeba používat. V této práci je třeba využití modulů SMTP, DHCP, DHCPs, DNS a ICMP (používán pro zjišťování odezvy pomocí příkazového řádku a příkazu ping). Moduly pro funkci nižších vrstev jako TCP, IP, ARP, MAC a jiné další (moduly aplikační vrstvy) jsou vkládány automaticky, takže se uživatel nemusí starat o to, co je třeba udělat pro použití výše zmíněných modulů. Stack je poměrně robustní, co se týče vkládání modulů, protože i kdyby uživatel nevložil moduly potřebné pro práci těch, které využívá, automaticky se vloží vše potřebné.

Dále je v modulu TCPIPConfig.h možno nakonfigurovat jméno, MAC adresu a IP adresu zařízení, maximální možný počet obsluhovaných TCP socketů a jiné další záležitosti týkající se parametrů některých protokolů stacku. Důležité je nakonfigurovat délku příchozích a odchozích datových rámců, jestliže provozujeme ve své aplikaci TCP server, protože počátečně nastavená velikost těchto rámců je příliš malá.

5.1.2 Vytvoření aplikace s Microchip TCP/IP stack

Níže uvedený kód představuje základní aplikaci s Microchip TCP/IP stackem. Aby fungovala správně, je třeba do obslužné rutiny přerušení vložit funkci TickUpdate, která využívá přerušovací flag od časovače mikrokontroléru Timer0. Celý stack pracuje na principu kooperativního multitaskingu, kdy jsou jednotlivé části kódu každé úlohy zpracovávány po úsecích, jejichž délka není stejná. Po dobu zpracování tohoto úseku nevyužívá mikrokontrolér žádná jiná úloha ani nadřazený operační systém. Tím je myšleno to, že se zpracuje vždy jistý úsek kódu úlohy a uchovají se potřebná data a pozice, na které se zpracovávání úlohy zastavilo. Dále se pokračuje na další úlohu, kde se děje to samé. Až po zpracování aktuálních úseků ve všech úlohách se dojde opět k první úloze, kde se pokračuje s dříve uchovanými daty a od pozice kde se skončilo. Takovéto úlohy lze vytvořit za pomoci stavového automatu, kde pozice(stav automatu), kterou je třeba uchovat a potřebná data realizujeme za pomoci statických lokálních proměnných. Na tomto principu pracuje většina modulů Microchip TCP/IP stacku.

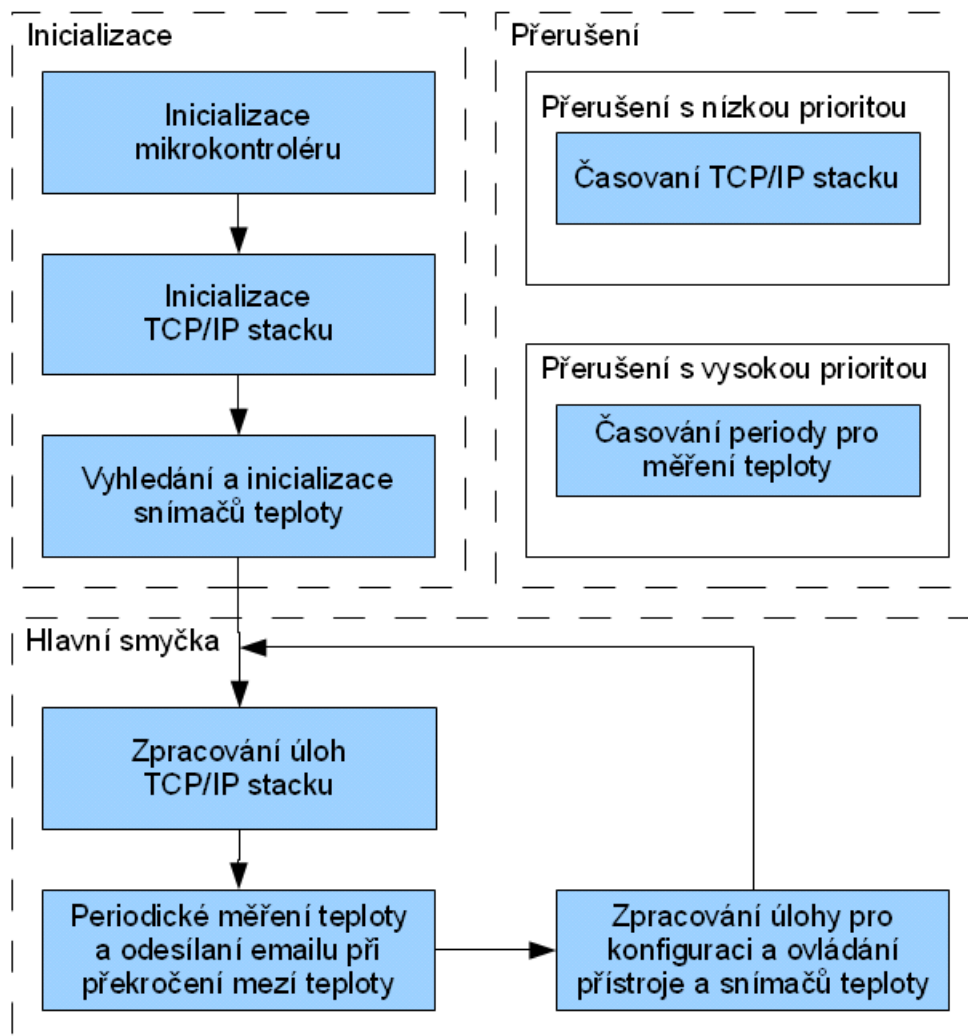
```
void main(void)
{
    TickInit();
    InitAppConfig();
    StackInit();

    while(1)
    {
        StackTask();
        StackApplications();
    }
}
```

V uvedeném kódu je možno vidět funkci TickInit, která slouží pro inicializaci časovače Timer0 mikrokontroléru. Dále je zde funkce InitAppConfig, ve které je inicializována struktura APP_CONFIG. Tato je důležitou součástí stacku, protože se v ní nachází, některé dříve uvedené konfigurační parametry. Ve funkci StackInit dochází k inicializaci modulů zajišťujících funkci všech vrstev stacku, od těch nejnižších až po ty ve vrstvě aplikační. Hlavní smyčka obsahuje funkci StackTask, která obsahuje obslužné úlohy pro všechny vrstvy stacku od transportní dolu. Konečně je zde funkce StackApplications v níž se nachází úlohy pro všechny protokoly aplikační vrstvy.

5.1.3 Popis hlavního programu

Na níže uvedeném obrázku 5.2 je znázorněn vývojový diagram hlavního programu. Obrázek je rozdělen do tří sekcí. První obsahuje sekvenci kódu zajišťující inicializaci, druhá ukazuje obslužné rutiny přerušení a třetí je hlavní smyčka programu, ve které se cyklicky vykonávají jednotlivé úlohy programu.



Obr. 5.2: Hlavní program

Inicializace

Inicializace probíhá ve třech krocích jak je znázorněno na obrázku. Jako první je provedena inicializace mikrokontroléru, kde se provede nastavení využívaných periférií. Jedná se především o výběr zdroje hodinového signálu a jeho nastavení, povolení přerušení a systému priorit pro přerušení. Dále je provedena inicializace I2C. Ta

spočívá ve výběru MSSP periferie (PIC18F97J60 obsahuje dvě) a její konfiguraci. Konečně je inicializován časovač Timer1 pro časování periody měření snímačů teploty.

Inicializace TCP/IP stacku byla krátce objasněna v předchozí podkapitole o vytváření aplikace s Microchip TCP/IP stackem a proto se na tomto místě již nebude znovu rozebírat.

Poslední ze tří bloků inicializace podle výše uvedeného obrázku, představuje vyhledání a inicializaci snímačů teploty. Proces vyhledání snímačů byl zatím jen naznačen v jedné z předchozích kapitol v souvislosti s objasňováním principů komunikace po rozhraní 1-Wire. Teprve v následující podkapitole bude objasněn detailně. Po tom co jsou snímače teploty nalezeny je provedeno převedení dat z jejich E2PROM paměti do SCRATCHPAD souboru registrů. Konkrétně se jedná o převedení údajů o mezích teploty pro spouštění alarmu a o převedení dat pro konfigurační registr.

Přerušení

Na mnoha periferiích mikrokontroléru může být jistou událostí vyvoláno přerušení. PIC18F97J60 má schopnost přiřadit těmto přerušení jednu ze dvou úrovní priorit a to buď nízkou a nebo vysokou. Každá z těchto úrovní je reprezentována svým vektorem přerušení, který se nachází pro vysokou prioritu na nižší adrese a pro nízkou na vyšší adrese ze začátku paměťového prostoru mikrokontroléru. Z těchto se potom skáče na odpovídající obslužné rutiny. Zdroj přerušení je rozeznán podle nastavení přerušovacího flagu periferie, která přerušení vyvolala.

Na obrázku je znázorněno, že v obslužné rutině s nízkou úrovní přerušení je prováděno časování pro TCP/IP stack. Toto přerušení je periodicky generováno časovačem Timer0 mikrokontroléru. Jak již bylo dříve zmíněno, je třeba v tomto přerušení volat funkci TickUpdate, která zajistí vše potřebné, tedy inkrementaci vnitřní proměnné TCP/IP stacku dwInternalTicks a vynulování přerušovacího flagu.

V obslužné rutině s vysokou úrovní přerušení je prováděno časování periody měření pro snímače teploty. Pro časování této periody je využit časovač Timer1 mikrokontroléru. Od začátku bylo plánováno využít jako zdroj hodinového signálu pro tento časovač sekundární oscilátor s kmitočtem 31,786kHz. Po tom co byl Timer1 nakonfigurován, tak aby měl jako zdroj hodinového signálu tento oscilátor, bylo zjištěno, že tento oscilátor nefunguje. Přesný důvod nebyl z nedostatku času a zdrojů určen. Proto byl pro Timer1 zvolen jako zdroj hodinového signálu vnitřní kmitočet mikrokontroléru (podělený čtyřmi), který se získal z primárního oscilátoru ve spolupráci s PLL. Pomocí tohoto kmitočtu (10,461675MHz) není možno dosáhnout toho, aby čítač vyvolal přerušení vždy jednou za sekundu (což je možno s kmitočtem 31,786kHz), proto je třeba vypočítat počet přerušení za jednu sekundu.

Doba jednoho přerušení:

$$t = \frac{1}{F_{osc}/4} * T1CountRegMax = \frac{1}{10416675Hz} * 65536 = 0,0063s \quad (5.1)$$

kde F_{osc} je vnitřní kmitočet mikrokontroléru a $T1CountRegMax$ je maximální počet hodnot 16-bitového čítacího registru pro Timer1.

Počet přerušení za 1s:

$$i = \frac{1}{t} = \frac{1}{0,0063} = 159 \quad (5.2)$$

Napočítáním 159 přerušení je nyní možno provádět měření teploty v časových intervalech, které jsou násobkem jedné sekundy (násobkem 159 přerušení od časovače Timer1).

Hlavní smyčka

Jednotlivé bloky uvnitř hlavní smyčky na výše uvedeném obrázku představují úlohy, které jsou cyklicky vykonávány mikrokontrolérem. Zpracování úloh TCP/IP stacku bylo uvedeno dříve. Popis úlohy starající se o periodické měření teploty a odesílání emailu stejně jako úlohy pro konfiguraci a ovládání přístroje a snímačů teploty bude uveden v samostatných podkapitolách.

5.2 Komunikace se snímači teploty

V této části textu budou popsány principy komunikace mezi snímači teploty a mikrokontrolérem. Bude zde objasněn způsob vyhledávání ROM identifikačního kódu a jeho využití k adresování snímačů.

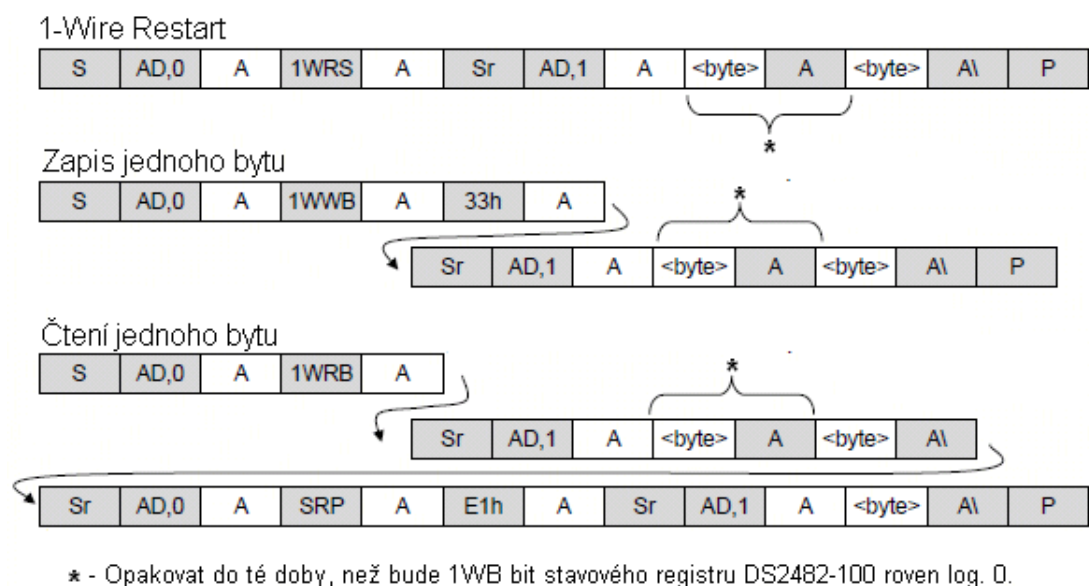
5.2.1 Rutiny pro rozhraní I2C a 1-Wire

Na to aby bylo možno komunikovat se snímači teploty, je třeba rutin pro dvě různá komunikačních rozhraní. V první řadě se jedná o rutiny pro rozhraní I2C a v druhé řadě jde o rutiny pro rozhraní 1-Wire. Na základě rutin pro I2C je potom možno napsat ty pro 1-Wire, protože jak již bylo zmíněno, pro generování sekvencí signálů na 1-Wire rozhraní, je využito převodníku DS2482-100, který je pomocí I2C ovládán.

Rutiny pro 1-Wire

Jako první jsou popsány rutiny pro 1-Wire. To z toho důvodu, že v dokumentaci k integrovanému obvodu DS2482-100 je návod, který popisuje jak tyto rutiny napsat. Dle tohoto návodu je vhodné přizpůsobit I2C rutiny tak, aby jednoduše sedly k uváděnému návodu. Proto je vhodné zaměřit se nejdříve na ty pro 1-Wire. Zde

nebudou popsány všechny, protože by to značně navyšovalo celkový obsah tohoto dokumentu. Níže budou uvedeny pouze tři hlavní. Jedná se o rutiny pro čtení a zápis jednoho bytu na rozhraní 1-Wire a provedení restart sekvence.



Obr. 5.3: I2C sekvence pro vykonání restartu, zápisu a čtení na 1-Wire rozhraní [6]

Na obrázku 5.3 je úplně nahoře uvedena sekvence operací na I2C tak, aby to provedlo restart na 1-Wire rozhraní. Šedá políčka vyjadřují směr komunikace master-slave, bílá opačně slave-master. Políčko S představuje startovní sekvenci. AD,0 adresuje DS2482-100 a zároveň mu dává vědět, že je vybrán pro zápis. DS2482-100 vysílá potvrzovací sekvenci A. Master následně vysílá příkaz pro restart na 1-wire sběrnici (příkaz je uveden v kapitole popisující DS2482-100). DS2482-100 opět potvrzuje a následně je masterem vysílána sekvence opětovného staru, po které znovu přichází adresace DS2482-100, ale tentokrát master dává vědět, že bude číst. DS2482-100 potvrdí, a vysílá data o velikosti jednoho bytu, ve kterých se nachází aktuální obsah stavového registru. To se opakuje tak dlouho, než se změní hodnota 1WB bitu ve stavovém registru z log. 1 na log. 0. V případě, že se tak stane, je masterem vysílána nepotvrzovací sekvence a konečně sekvence zakončení přenosu na I2C.

V prostřední části obrázku je uvedeno, jak provést zápis jednoho bytu na rozhraní 1-Wire. Je zde pouze rozdíl v tom, že se vyšle jiný příkaz a následně požadovaná data (33h), zbytek je stejný jako u restartu.

Nakonec je na obrázku znázorněno čtení jednoho bytu z 1-Wire rozhraní. Ze začátku je možno vidět opět velice podobnou posloupnost sekvencí a čekání než se dokončí čtení se změnou 1WB. Po tom co se čtení dokončí je třeba provést opětovnou adresaci s oznámením toho, že bude do DS2482-100 zapisováno. Po dokončení tohoto oznámení je třeba vyslat instrukci pro nastavení ukazatele čtení a následně i označení

registru, ze kterého bude čteno. Tímto registrem je registr pro čtení přijatých dat z rozhraní 1-Wire. Po tom co je toto nastavení provedeno master znovu provádí adresování s požadavkem na čtení ze slave. Konečně jsou přenesena data, která byla přečtena za pomoci DS2482-100 z ozhraní 1-Wire a je ukončeno spojení.

Rutiny pro I2C

Vzhledem k výše uvedenému rozboru rutin pro 1-Wire, lze nyní sepsat rutiny pro I2C, tak aby každému rámečku na obrázku 5.3 odpovídala jedna funkce v programu. Bude tím zajištěna dostatečná flexibilita pro nepatrné rozdíly mezi třemi výše uvedenými příklady a ušetří se paměť programu.

Jednotlivé rámečky jsou v souladu s časovým průběhem signálu na rozhraní I2C a MSSP periferie mikrokontroléru je dokáže generovat s pomocí ovládání registrů k tomu určených. Výsledné rutiny potom v případě rámečků S, Sr, P, A a /A představují pouze nastavení příslušných bitů v kontrolním registru MSSP periferie a čekání na dokončení operace automatickým vynulováním tohoto bitu. Zápis a čtení je prováděno podobně. Je třeba pouze navíc pracovat s bufferem pro data a jemu náležícímu kontrolnímu bitu v kontrolním registru. Při používání zápisu jako adresace je dále třeba hlídat nejméně významný bit adresy, který určí zda-li bude dále při komunikaci master ze slave číst nebo do něj bude zapisovat.

Nyní již máme vše potřebné pro sestrojení následujících funkcí, tedy vyhledávání a adresaci 1-Wire zařízení, které jsou pro práci se snímači DS18B20 klíčové.

5.2.2 Vyhledávání a adresace 1-Wire zařízení

Pořídíme-li si nějaké 1-Wire zařízení, jako zde využívaný teploměr nebo například digitální potenciometr, paměť EEPROM a jiné, pravděpodobně nedostaneme přímo jeho ROM identifikační číslo. V případě, že budeme na 1-Wire sběrnici využívat více než jedno zařízení, bude třeba identifikační čísla nějakým způsobem zjistit. Je možné připojovat jednotlivá zařízení postupně a zjišťovat tak jejich identifikační číslo pomocí dříve uvedeného příkazu čtení ROM pouze jednoho připojeného zařízení na sběrnici. Jednotlivá identifikační čísla by se postupně ukládala do paměti a následně by je bylo možno využít pro adresaci. Tento způsob zjišťování identifikačních čísel není příliš šikovný. Mnohem lepší způsob identifikace všech zařízení se nabízí využitím vyhledávacího příkazu.

1-Wire vyhledávací algoritmus

Každé vyhledávání na 1-Wire sběrnici začíná restartem rozhraní a vysláním vyhledávacího příkazu. Po tom, co zařízení obdrží tento příkaz, začnou všechna současně

vysílat svůj identifikační kód nejméně významným bitem napřed. Dejme tomu, že zařízení odeslala první bit. Současně se nyní na sběrnici mohou nacházet dvě různé hodnoty tohoto bitu. Některá zařízení vyslala log. 1 a některá log. 0. Vzhledem k tomu, že 1-Wire sběrnice pracuje na principu otevřeného kolektoru, bude tento bit masterem čten jako výsledek operace logického součinu prvních bitů vyslaných od všech připojených zařízení. Aby bylo možno zjistit jaké hodnoty jsou ve vysílaném bitu, je hned po tomto odesílání ještě jeho negace. Odesílání tohoto negovaného bitu je důležité pro zjištění skutečné hodnoty identifikovaného bitu. Následující tabulka ukazuje čtyři možnosti které mohou nastat.

ROM bit	Negace ROM bitu	Popis
0	0	Na identifikované pozici jsou dvě možné log. hodnoty jednoho bitu.
0	1	Na identifikované pozici jsou pouze log. nuly.
1	0	Na identifikované pozici jsou pouze log. jedničky.
1	1	Na sběrnici nejsou zařízení.

Tab. 5.1: Možné stavy na 1-Wire sběrnici při vyhledávání ROM kódu[15]

Po tom, co master přijme tento první bit, je třeba aby rozhodl, která zařízení budou pokračovat ve vyhledávání, a která budou odpojena. To se provede tak, že master odešle na sběrnici jeden bit a jestli-že je jeho hodnota souhlasná s hodnotou, kterou zařízení odeslalo jako první bit, tak zařízení zůstává. V opačném případě se zařízení odpojí a čeká na příkaz 1-Wire restart a případné další instrukce. Rozhodnutí o tom, které zařízení zůstává a které bude odpojeno, je provedeno vždy pro první případ z tabulky a je třeba si během vyhledávání pamatovat vždy poslední pozici, kde došlo k tomuto rozhodování. To proto, aby se při následujícím vyhledávání identifikačního čísla, odpojilo na této pozici to zařízení, u kterého je jeho identifikační číslo již zjištěno. Jakmile se tedy znovu dostaneme na tuto pozici je třeba masterem odeslat opačnou hodnotu jednoho bitu než dříve. Co se týče rozhodování u druhého a třetího řádku tabulky, tak se masterem odesílá vždy stejná hodnota jako právě zjištěná. Odeslání opačné hodnoty by způsobilo odpojení všech zařízení. V posledním případě je odesílání čehokoliv zbytečné. Nicméně je možno takto zjistit, že bylo fyzicky přerušeno vedení.

Uvedený postup se opakuje pro ostatních 63 bitů. Během toho se identifikuje jedno zařízení. Po ukončení identifikace jednoho zařízení se pokračuje na další tak, že se provede restart na 1-Wire sběrnici, odešle se vyhledávací příkaz a projde se opět celý 64 bitový identifikační kód. Více informací je možno nalézt v [15].

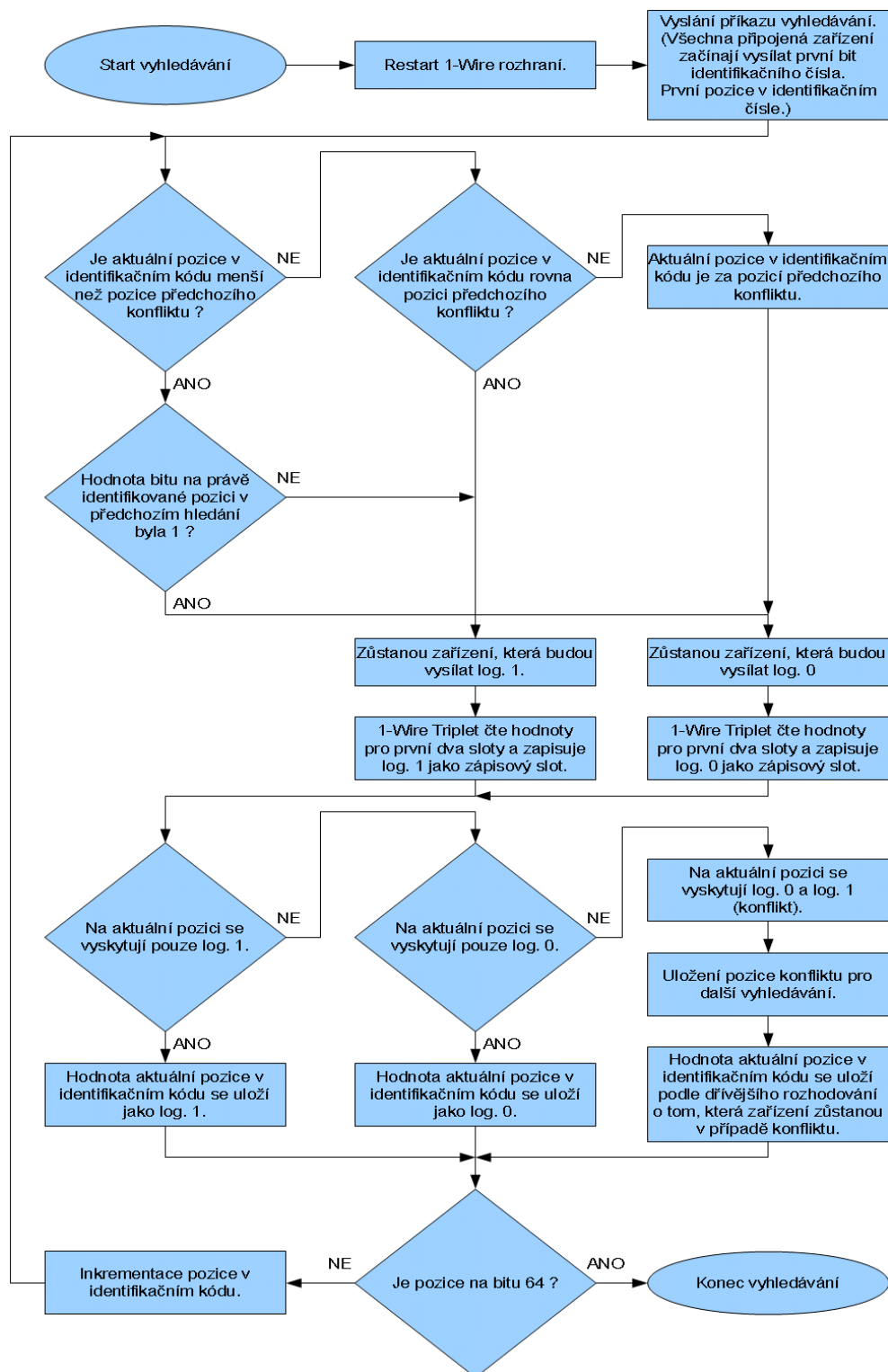
Vyhledávání za pomoci příkazu 1-Wire Triplet

V souvislosti z předchozím popisem integrovaného obvodu DS2482-100 byl krátce popsán příkaz 1-Wire Triplet. Tento příkaz je schopen generovat na jedno použití tři časové sloty na 1-Wire rozhraní. Jedná se o dva sloty pro čtení (Obr.3.10) a jeden pro zápis (Obr.3.9). První slot pro čtení pojme aktuálně vysílaný bit od zařízení zúčastněných vyhledávání a druhý pojme jeho negaci, jak bylo dříve zmíněno. Tato dvojice čtecích slotů je naplněna tedy jednou z možných kombinací uvedených v tabulce 5.1. Do zapisovacího slotu bude zapsána hodnota informující zařízení o tom, která mají zůstat v právě probíhajícím vyhledávání. Tato hodnota je využita příkazem jenom v případě, že jsou právě čtené dva sloty o hodnotě log. 0 (konflikt). V případě, že konflikt nenastane, je automaticky generována pro zápis hodnota stejná jako ta, kterou vysílala všechna připojená zařízení. Důležité je, aby hodnota pro zápisový slot, tedy rozhodnutí o tom, které ze zařízení budou pokračovat, byla známa dříve než bude zápis proveden. Nelze se rozhodnout podle právě čtených pozic, ale k nim odpovídající hodnota zápisového slotu musí podle nich určit, jaká zařízení zůstanou a jaká budou odpojena. To proto, že všechny tři sloty jsou zpracovány naráz a nelze znát hodnotu čtených bitů v prvních dvou slotech ještě před tím, než se provede slot třetí. Algoritmus, který realizuje vyhledávání a využívá k tomu tento příkaz, musí být tedy sestaven tak, aby toto ošetřoval. Na obrázku 5.4 je tento algoritmus znázorněn.

Ještě je třeba podotknout, že mezi koncem jednoho vyhledávání a začátkem dalšího je třeba uchovávat vždy poslední nalezený identifikační kód. Na obrázku 5.4 je možno vidět, že takto uchovaný předchozí kód je využíván pro rozhodnutí o tom, která zařízení zůstanou v právě probíhajícím vyhledávání v případě, že už byl v dřívějším hledání nalezen nějaký identifikační kód. V takovém případě se opisují dřívější hodnoty identifikačního kódu do té doby, než se narazí na pozici konfliktu. Na pozici konfliktu je třeba zapsat jinou hodnotu než tomu bylo v předchozím hledání, jinak by byl nově nalezený identifikační kód stejný jako ten předchozí. Z toho plyne, že je mezi jednotlivými hledáními třeba uchovávat ještě pozici posledního konfliktu.[6]

Adresace 1-Wire zařízení

V případě, že již známe identifikační čísla všech zařízení připojených na 1-Wire rozhraní, můžeme provádět jejich adresaci. Ta je podstatně jednodušší než vyhledávání ROM kódů. V první řadě se provede opět úvodní sekvence složená z restartu 1-Wire rozhraní. Po tom se vyšle příkaz pro porovnání ROM kódu a za ním následuje jeho vysílání. Zařízení na rozhraní 1-Wire čtou ROM kód po jednotlivých bitech a v případě, že se právě vysílaný bit neshoduje s odpovídajícím bitem jejich ROM kódu,



Obr. 5.4: Vývojový diagram pro vyhledávání 1-Wire zařízení za pomoci příkazu 1-Wire Triplet

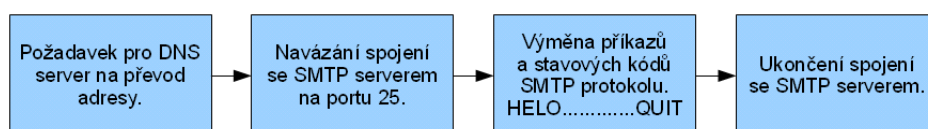
dojde k jejich odpojení. Odpojená zařízení potom čekají, než přijde další restart rozhraní. To zařízení, které projde tímto procesem až do konce, bude následně přijímat další příkazy.

5.3 Odesílání emailu při překročení určených mezí teploty

Níže uvedený text uvede čtenáře do problematiky protokolu SMTP. Dále objasní algoritmus, který obstarává periodické měření teploty a odesílání emailu v případě, že některý ze snímačů překročí jemu povolený rozsah teplot.

5.3.1 Odeslání emailu a SMTP protokol

SMTP protokol, stejně jako každý protokol aplikační vrstvy, využívá pro svoji funkci protokolů nižších vrstev. Přesněji řečeno využívá TCP spojení se vzdáleným serverem. Spojení se SMTP serverem je standardně prováděno na portu 25. V některých případech je toho spojení uskutečněno i na jiných portech. To se děje v případě, chceme-li využít šifrované TCP spojení za pomoci protokolu SSL. Pro TCP spojení je potom namísto portu 25 využíván port 465. Aby bylo možno navázat spojení na některý z těchto portů, je třeba nejdříve požádat DNS server o převedení adresy SMTP serveru ve formátu řetězce(doména) na IP adresu. Takto uvedený postup je znázorněn blokově na obrázku 5.5 v prvních dvou blocích.



Obr. 5.5: Posloupnost operací při odesílání emailu

Podarí-li se vytvořit toto spojení, může se začít komunikovat se SMTP serverem (třetí blok obrázku 5.5). Tato komunikace probíhá za pomoci sady standardizovaných příkazů a stavových kódů. Příkazy jsou vysílány SMTP klientem a jsou reprezentovány vždy čtyřmi znaky. Stavové kódy jsou reakcí SMTP serveru na příkazy od klienta a jsou ve formátu tří číslic. Příkazy mohou obsahovat parametr, který je oddělen mezerou. Podobně stavové kódy mohou být následovány zprávou, která je zase oddělena mezerou. Příkazy a stavové kódy jsou ukončovány dvojicí znaků pro odřádkování (CRLF). Na níže uvedeném textu je příklad výměny příkazů a stavových kódů mezi klientem a serverem. Nyní předpokládejme jako výchozí stav situaci,

kdy již bylo navázáno připojení na SMTP server. Pro tento systém na monitorování teploty je využíváno SMTP serveru portálu Seznam. Tento vyžaduje autentizovaný přístup pro odeslání emailu a proto je třeba vytvořit na tomto portálu emailový účet.

```
S: 220 2.0.0 Seznam SMTP server waiting for your HELO/EHLO
C: EHLO smt
S: 250-Email.Seznam.cz - Email zdarma na cely zivot ESMTP
S: 250-AUTH LOGIN PLAIN
S: 250-8BITTIME
S: 250-PIPELINING
S: 250-SIZE 18000000
S: 250-ENHANCEDSTATUSCODES
S: 250 X-SZNEXTENSIONS
C: AUTH LOGIN
S: 334 VXN1cm5hbWU6
C: bW9uaXRvcnRlcGxvdHlAc2V6bmFtLmN6
S: 334 UGFzc3dvcmQ6
C: dGVwbG90eW1vbml0b3I=
S: 235 2.2.0 Authentication succeeded
C: MAIL FROM:<monitorteploty@seznam.cz>
S: 250 2.1.0 Ok <monitorteploty@seznam.cz>
C: RCPT TO:<xpapez03@stud.feec.vutbr.cz>
S: 250 2.1.5 Ok <xpapez03@stud.feec.vutbr.cz>
C: DATA
S: 354 Enter message, ending with <CRLF>.<CRLF>
C: From:"System na monitorovani teploty"<monitorteploty@seznam.cz>
  To:"Milan Papez"<xpapez03@stud.feec.vutbr.cz>
  Subject:Prekrojeni povoleného rozsahu teploty

  Byl překročen povolený interval teplot. Jedna se o snimace: 1,2.
  .
S: 250 2.0.0 Mail xxxxxxxxxx queued for delivery in session xxxxxxxxxxxxxx.
C: QUIT
```

První řádek ukazuje, že po otevření spojení server odpovídá stavovým kódem 220 a uvítacím pozdravem, čímž dává najevo, že je připraven pro komunikaci. Klient následně odešle příkaz EHLO následovaný parametrem, který představuje hostitelské jméno. Mimochodem, v uvítací zprávě od serveru je možno vidět, že tento příkaz nabízí spolu s dalším příkazem HELO. HELO je původně využívaný příkaz v základní

verzi protokolu SMTP. Časem byl totiž tento protokol rozšířen na verzi ESMTP, kde byl tento příkaz změněn prohozením prvních dvou písmen. Rozšíření s sebou přineslo ještě další změny, mezi něž patří rozšíření délky zprávy, rozšíření příkazů, stavových slov apod. Po přijetí příkazu EHLO server vyšle opakovaně stavový kód 250 následovaný několika řetězci. Stavový kód 250 je využíván jako informace pro klienta o tom, že jeho příkaz byl i s parametrem úspěšně přijat. Zde je možno vidět, že je použit pro vypsání několika informací, ze kterých je zde nejdůležitější AUTH LOGIN PLAIN. Tato informace říká, že pro odeslání emailu je třeba autentizace a je tedy třeba vlastnit emailový účet, jak bylo již dříve řečeno. Aby bylo možno provést autentizaci musí klient zaslat příkaz AUTH následovaný parametrem LOGIN. Na to mu server odpoví stavovým kódem 334 následovaným slovem Username:, které je zakódováno jako Base64 řetězec.

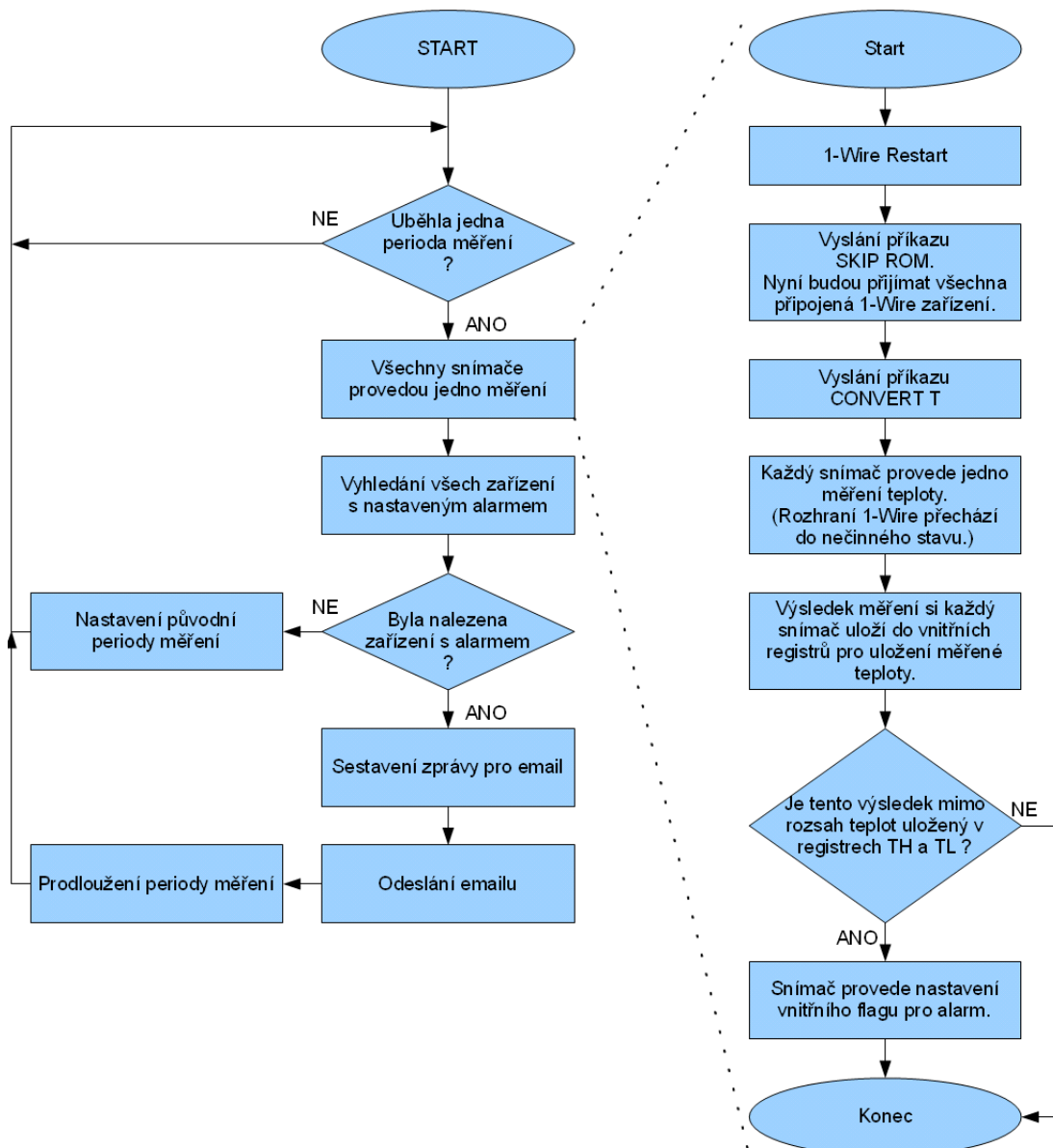
Na tomto místě není příliš účelné objasňovat jak funguje kódování Base64 a proto v případě zájmu čtenář dohledá informace sám.

Po zadání uživatelského jména a hesla server odpoví stavovým kódem 235 a řetězcem, který zároveň objasňuje, co dané stavové slovo znamená. Po úspěšném přihlášení může klient zapsat od koho bude email odeslán, což se provede zasláním příkazu MAIL, za nímž následuje parametr FROM: s emailovou adresou odesílatele. Server to opět potvrdí stavovým kódem 250. Podobně to probíhá i při zadávání příjemce. Po tom, co je příjemce potvrzen, klient vyšle příkaz DATA na což server odpoví kódem 354, který potvrzuje a oznamuje, že zprávu je třeba ukončit tečkou z obou stran oddělenou znaky CRLF. Následně může klient začít vysílat vlastní zprávu. Je vidět, že se v ní mohou nacházet data, která jsou obvykle při prohlížení pošty vidět nad přijatou zprávou (From, To, Subject). Tyto by měla být vždy odřádkována. Mezi nimi a tělem zprávy je třeba odřádkovat dvakrát. Ukončení je provedeno jak již bylo řečeno pomocí tečky z obou stran oddělenou znaky CRLF. Po oddělení server potvrdí a klient ukončuje spojení příkazem QUIT.

5.3.2 Algoritmus pro periodické měření teploty a odeslání emailu při překročení povolených mezí teplot

Algoritmus pro odeslání emailu, při překročení přednastavených mezí teploty, ukazuje obrázek 5.6. Tento je sestaven tak, že se v nekonečné smyčce hlídá, zda uběhl čas pro jednu periodu měření (dříve uvedené přerušení od Timer1). Uběhne-li jedna perioda, všechny snímače zároveň provedou jedno měření teploty. Pro objasnění, co se děje na rozhraní 1-Wire během tohoto měření, je tento blok vývojového diagramu na obrázku 5.6 rozkreslen.

Aby bylo možno provést měření u všech snímačů zároveň, je třeba použít příkazu pro přeskočení ROM identifikačního kódu. Tím je vynechán proces adresace jednot-



Obr. 5.6: Vývojový diagram pro periodické měření teploty a odesílání emailu

livých zařízení a postupného přikazování měření pro každý snímač zvlášť. Zároveň je tím proces měření velice zkrácen. Po tom co je toto měření provedeno si každý snímač uloží výsledek do vnitřních registrů k tomu určených. Po tom, co je teplota uložena, každý snímač provede porovnání těchto registrů s registry pro určení povoleného intervalu teploty (TH a TL obrázek 3.13). V případě překročení intervalu se nastaví interní flag pro alarm.

Následující krok využívá dříve zmíněného algoritmu pro vyhledávání 1-Wire zařízení, pouze s tím rozdílem, že je využit příkaz pro vyhledání jen těch zařízení, která mají nastavena alarm. Jsou-li nalezeny snímače, které překročily rozsah povolených

teplot, je sestavena zpráva podle dříve uvedeného popisu. Tato zpráva obsahuje informaci o tom, že byl rozsah překročen a také čísla snímačů, kterých se to týká.

Po odeslání emailu, je prodloužena perioda měření, aby nedocházelo k příliš častému odesílání v případě, že se teplota navýšila na delší dobu. Po uběhnutí této delší periody se v případě normálního stavu teploty vrátí perioda měření k normálu, v opačném případě se opět prodlouží.

5.4 Příkazy pro konfiguraci přístroje a měření teploty

Důvodem pro vytvoření možnosti konfigurovat přístroj jsou zcela jasné. Jednak je vhodné, aby bylo možno měnit adresu příjemce emailu aniž by musel být znovu nahrazen firmware mikrokontroléru a také je třeba mít možnost změnit další parametry, jako periodu měření, určit meze teploty pro každý snímač apod. Dále je vhodné vytvořit příkazy na čtení dat ze snímačů teploty, za pomoci kterých, by bylo možné vytvořit program určený pro zobrazení a záznam měřených dat.

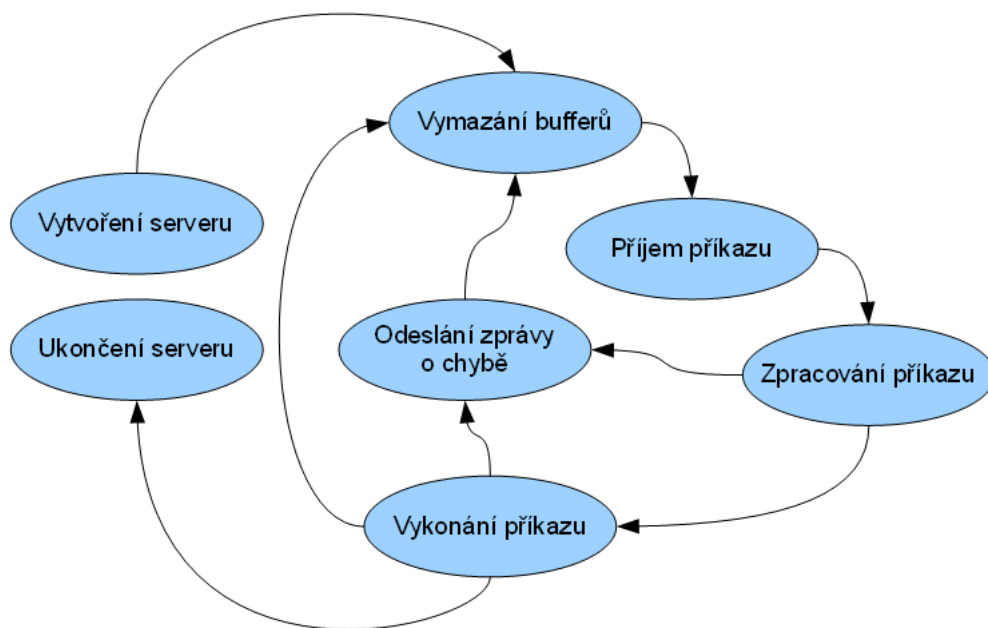
Příkazy se tedy celkově dělí do dvou kategorií. První obsahuje příkazy pro konfiguraci přístroje. Ve druhé jsou příkazy pro měření teploty a konfiguraci snímačů. Příkazy pro konfiguraci přístroje jsou uvedeny v tabulce 5.3, příkazy pro měření teploty v tabulce 5.4.

5.4.1 Úloha pro zpracování příkazů

Celkový vývojový diagram pro zpracování všech příkazů by byl rozsáhlý, protože každý z příkazů vyžaduje specifické zpracování jeho parametrů, proto je na obrázku 5.7 uveden pouze stavový diagram úlohy.

Objasnění bude provedeno na příkladu zpracování jednoho z poskytovaných příkazů, kterým je příkaz čtení nastavených mezí pro alarm snímače (ralm). Výměna příkazů pracuje na principu TCP spojení mezi klientem a serverem, přičemž samotný přístroj zastává úlohu serveru. Klientem může být vytvořený program využívající připojení na tento server nebo například program Microsoft HyperTerminál.

Po zapnutí přístroje se provede vytvoření serveru, který naslouchá na portu 9760. Po tom, co se server vytvoří, dojde k vymazání obsahu bufferů. Jedná se o buffer pro příkaz, pro parametry příkazu a pro dočasné ukládání. Jsou-li tyto vymazány, přejde se do stavu pro příjem příkazu. V případě, že se někdo připojí a začne posílat jednotlivé znaky, budou tyto přijímány a okamžitě vráceny klientovi jako zpětná vazba toho, že spojení probíhá. Níže uvedený příklad ukazuje zadání příkazu pro čtení nastavených mezí alarmu u snímače číslo 2.



Obr. 5.7: Stavový diagram úlohy pro zpracování příkazů

C: ralm.2
 S: 25
 10

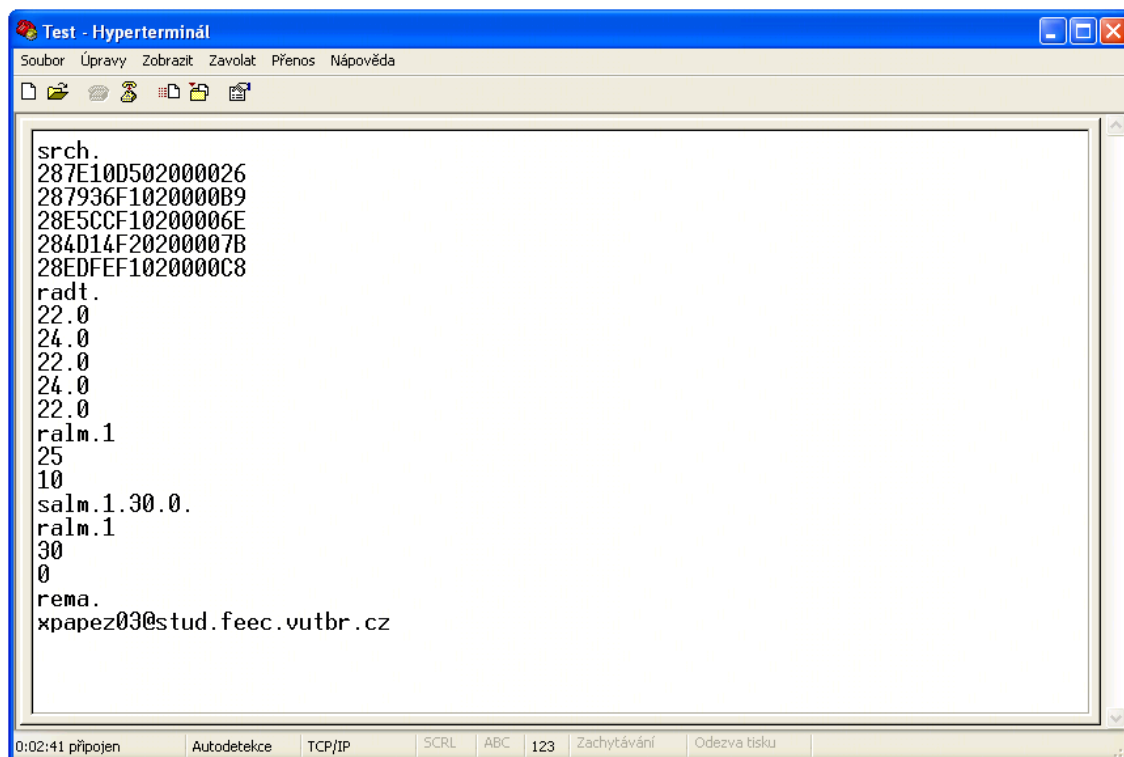
Po té co server zachytí příkaz pro odřádkování, přejde do stavu zpracování příkazu. V tomto stavu se postará o rozdělení příkazu a jeho argumentu do dříve zmíněných bufferů. Nová data v bufferu pro příkaz se porovnají se sadou známých příkazů a najde-li se ten odpovídající, je uloženo jeho číslo pro nadcházející stav, kde je vykonán. Ještě před tím se zde provádí nejzákladnější kontrola toho, zda-li je příkaz ve formátu, ve kterém má být. Kontroluje se zda-li je na páté pozici přijatého řetězce oddělovací znak tečky. V případě, že tomu tak není, vyhodnotí se to jako chyba, uloží se číslo chyby a přejde se do dalšího stavu pro odeslání chybového hlášení. V tomto se s pomocí čísla chyby určí řetězec identifikující chybu a je vyslán směrem ke klientovi. Následně se přejde do stavu pro vymazání bufferů a dále do stavu pro příjem příkazu, kde se čeká na další příkaz.

Je-li oddělovací znak zadán správně, přejde se do stavu vykonání příkazu. Zde se podle dříve zmíněného čísla příkazu určí, o který příkaz se jedná a provede se odpovídající zpracování parametrů. Jsou-li parametry zadány špatně, dojde k identifikaci chyby, určení jejího čísla a přejde se opět do stavu pro odeslání chybového hlášení, kde se děje to samé jak bylo řečeno dříve, pouze se odešle jiné chybové hlášení. Budou-li parametry zadány dobře, provede se činnost spojená s příkazem. V tomto příkladě se použije číslo za příkazem k volbě snímače teploty. Po přečtení

mezi teploty ze zvoleného snímače server odešle dva řetězce, které reprezentují horní a dolní mez alarmu. Dále se přejde do stavu vymazání bufferů a potom se opět čeká na příjem příkazu.

5.4.2 Ukázka komunikace s přístrojem

Obrázek 5.8 ukazuje příklad komunikace s přístrojem za pomoci programu HyperTerminál. Jako první je použit příkaz `srch`, který vyhledá zařízení připojená na rozhraní 1-Wire a následně vypíše jejich identifikační čísla. Dále je použit příkaz `radt`, který přečte a vypíše teplotu zase u každého připojeného zařízení. Příkaz `ralm.1` slouží k přečtení mezí teplot u snímače číslo 1. Příkaz `salm.1.30.0` provede nastavení těchto mezí. Nastavení je prováděno horní mezí napřed. Uvedený příklad tedy nadefinoval rozsah teplot pro alarm snímače v rozmezí 0 až 30°C. Dále je provedeno opětovné čtení, kde je vidět, že meze jsou skutečně nastaveny. Poslední uvedený příkaz slouží k přečtení emailové adresy příjemce emailu, který se odešle například právě tehdy, když u snímače jedna bude naměřeno 31°C.



```
Test - Hyperterminál
Soubor  Úpravy  Zobrazit  Zavolat  Přenos  Nápověda

srch.
287E10D502000026
287936F1020000B9
28E5CCF10200006E
284D14F20200007B
28EDFEF1020000C8
radt.
22.0
24.0
22.0
24.0
22.0
ralm.1
25
10
salm.1.30.0.
ralm.1
30
0
rema.
xpapez03@stud.feec.vutbr.cz

0:02:41 připojen  Autodetekce  TCP/IP  SCRL  ABC  123  Zachytávání  Odezva tisku
```

Obr. 5.8: Ukázka komunikace s přístrojem přes program HyperTerminál

Jednotky teploty jsou ve stupních Celsia, nicméně vzhledem k tomu, že se jedná o TCP komunikaci nejsou tyto přenášeny, aby se usnadnila případná realizace software pro zobrazování měřených dat.

Název:	Příkaz:	Popis:	Vstup:	Výstup:
Čtení IP adresy	ripa - Read IP Address	Přečte IP adresu zařízení.	-	IP adresa zařízení
Čtení defaultní IP adresy	rdip - Read Default IP Address	Přečte defaultní IP adresu zařízení.	-	Defaultní IP adresa zařízení
Čtení adresy email serveru	resa - Read Email Server Address	Přečte adresu aktuálně nastaveného SMTP serveru.	-	Adresa nastaveného SMTP serveru.
Čtení čísla portu email serveru	resp - Read Email Server Port	Přečte aktuálně nastavené číslo portu SMTP serveru.	-	Číslo portu u nastaveného SMTP serveru.
Čtení uživatelského jména	rusr - Read USeRname	Přečte aktuálně nastavené uživatelské jméno.	-	Nastavené uživatelské jméno.
Čtení uživatelského hesla	rpas - Read PASsword	Přečte aktuálně nastavené uživatelské heslo.	-	Nastavené uživatelské heslo.
Čtení email adresy příjemce hlášení	rema - Read EMail Address	Přečte aktuálně nastavenou email adresu příjemce hlášení.	-	Nastavená emailová adresa příjemce hlášení.
Čtení MAC adresy	rmac - Read MAC	Přečte aktuálně nastavenou MAC adresu.	-	MAC adresa přístroje.

Tab. 5.2: Příkazy pro konfiguraci přístroje - 1. část

Název:	Příkaz:	Popis:	Vstup:	Výstup:
Odešli email	seml - Send EMail	Odešle email na nastavenou adresu.	-	-
Nastavení adresy email serveru	sesa - Set Email Server Address	Nastaví novou adresu SMTP serveru.	Adresa SMTP serveru.	-
Nastavení čísla portu email serveru	semp - Set Email Server Port	Nastaví nové číslo portu SMTP serveru. (Není možno užívat SSL porty.)	Číslo portu pro SMTP server.	-
Nastavení uživatelského jména	susr - Set UserName	Nastaví uživatelské jméno pro autentizaci. (Musí být na SMTP serveru registrováno.)	Uživatelské jméno	-
Nastavení uživatelského hesla	spas - Set Password	Nastaví uživatelské heslo pro autentizaci.	Uživatelské heslo	-
Nastav emailovou adresu příjemce hlášení	sema - Set EMail Address	Nastaví adresu emailu příjemce hlášení.	Emailová adresa	-
Poslední chybové hlášení	lerm - Last Error Message	Vypíše poslední chybové hlášení.	-	Chybové hlášení

Tab. 5.3: Příkazy pro konfiguraci přístroje - 2. část

Název:	Příkaz:	Popis:	Vstup:	Výstup:
Vyhledání 1-Wire zařízení	srch - SeaRCH 1-wire devices	Vyhledá připojená 1-Wire zařízení.	-	Identifikační kód každého zařízení.
Přečti teplotu	rtmp - Read Temperature	Provede měření a čtení teploty zvoleného snímače.	Číslo snímače	Teplota
Přečti teplotu všech snímačů	rtas - Read Temperature of All Sensors	Provede postupné měření a čtení teploty u všech snímačů.	-	Teploty jednotlivých snímačů.
Přečti periodu měření	rpom - Read Period Of Measurement	Přečte periodu měření snímačů. Tato měří současně u všech snímačů.	-	Perioda měření v s.
Přečti způsob napájení	rpws - Read PoWer Supply	Přečte způsob napájení u vybraného snímače.	Číslo snímače	Způsob napájení
Přečti identifikační číslo	rsid - Read Sensor ID	Přečte identifikační ROM kódu vybraného snímače.	Číslo snímače	Identifikační kód zařízení.
Přečti meze pro alarm	ralm - Read ALarM	Přečte meze teplot pro alarm u vybraného snímače.	Číslo snímače	Meze teploty pro alarm. Horní mezí napřed.
Nastav periodu měření	spom - Set Period Of Measurement	Nastavení periody měření.	Perioda měření v sekundách. (120 - zadá dvě minuty)	-
Nastav meze pro alarm	salm - Set ALarM	Nastavení meze pro alarm u vybraného snímače.	Meze teploty pro alarm. Horní mezí napřed.	-

Tab. 5.4: Příkazy pro měření teploty, Pozn: Všechny příkazy a jejich parametry pracující s teplotou udávanou ve stupních Celsia.

6 ZÁVĚR

Výsledkem této práce je systém monitorování teploty pro serverovny a rozvodné uzly, který je schopen odeslat email při překročení nastavených mezí teploty. Tento email informuje příjemce o tom, které ze snímačů překročily povolený rozsah teplot. Zařízení je otestováno s připojením pěti snímačů na vzdálenosti deseti metrů. Délka jednoho segmentu je dva metry. Snímače umožňují měření teploty v rozsahu -75 až 125°C , přičemž přesnosti $0,5^{\circ}\text{C}$ dosahují v rozmezí -10 až 85°C . Základně nastavená perioda měření je 30 sekund. Výsledný systém je koncipován jako embedded systém bez zobrazovacích a ovládacích prvků. Jeho ovládání, konfigurace a získávání dat je prováděna vzdáleně s pomocí sady příkazů. Těmito je možno číst teplotu jednotlivých snímačů, nastavovat hlídané meze teploty pro každý snímač samostatně, měnit periodu měření, nastavit adresu příjemce emailu, číst parametry snímačů apod. Zpracování příkazů je vybaveno základní detekcí chyb při jejich zadávání.

Pro konstrukci zařízení bylo využito mikrokontroléru PIC18F97J60, který obsahuje ethernetový řadič. Pro měření teploty je využito snímačů DS18B20, které komunikují po rozhraní 1-Wire. Pro generování časových funkcí rozhraní 1-Wire bylo využito obvodu DS2482-100, který je řízen rozhraním I2C. Odeslání emailu je realizováno s pomocí protokolu SMTP, který je obsažen v TCP/IP stacku poskytovatelského výrobce mikrokontroléru. Hlídaní teploty je realizováno periodickým měřením a následným využitím 1-Wire funkce vyhledávání snímačů, které přejdou do stavu alarmu, jestliže překročí povolený teplotní rozsah. Pro zpracování příkazů běží v zařízení TCP server.

LITERATURA

- [1] HRBÁČEK, Jiří. *Komunikace mikrokontroléru s okolím - 1. díl*. Praha : BEN - technická literatura, 1999. 160 s.
- [2] HRBÁČEK, Jiří. *Komunikace mikrokontroléru s okolím - 2. díl*. Praha : BEN - technická literatura, 2000. 152 s.
- [3] DOSTÁLEK, Libor; KABELOVÁ, Alena. *Velký průvodce protokoly TCP/IP a systémem DNS*. Brno : Computer Press, a.s., 2008. 488 s.
- [4] MICROCHIP TECHNOLOGY INC. *PIC18F97J60 Family Data Sheet*. 2009. 480. [online]. [cit. 19.5.2011] Dostupné z WWW: <<http://ww1.microchip.com/downloads/en/DeviceDoc/39762f.pdf>>
- [5] MAXIM INTEGRATED PRODUCTS. *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*. 2002. 20. [online]. [cit. 19.5.2011] Dostupné z WWW: <<http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf>>
- [6] MAXIM INTEGRATED PRODUCTS. *DS2482-100 Single-Channel 1-Wire Master*. 2008. 22. [online]. [cit. 19.5.2011] Dostupné z WWW: <<http://datasheets.maxim-ic.com/en/ds/DS2482-100.pdf>>
- [7] *Autocont* [online]. 2010 Dostupné z WWW: <<http://muj.autocont.cz/images/reseni/datovacentra/horka-ulicka-01b.jpg>>.
- [8] *Autocont* [online]. 2010 Dostupné z WWW: <<http://muj.autocont.cz/images/reseni/datovacentra/reseni-in-row-chlazen-03b.jpg>>.
- [9] Jak a proč měřit teplotu serverů. *NET GURU* [online]. 2010. Dostupné z WWW: <<http://www.netguru.cz/network/jak-a-proc-merit-teplotu-serveru.html>>.
- [10] STMicroelectronics. *LD29150XXx 1.5 A very low drop voltage regulators*. 2008. 21. [online]. Dostupné z WWW: <<http://www.st.com/stonline/books/pdf/docs/9614.pdf>>
- [11] MICROCHIP TECHNOLOGY INC. *ENC28J60 Data Sheet* 2004. 102. [online]. [cit. 19.5.2011] Dostupné z WWW: <<http://ww1.microchip.com/downloads/en/DeviceDoc/39662b.pdf>>
- [12] *Freescale MQXTM Real-Time TCP/IP Communication Suite (RTCS)* [online]. 2010. [cit. 19.5.2011] Dostupné z WWW: <<http://www.freescale.com/webapp/sps/site/overview.jsp?code=MQXRTCS>>

- [13] *Microchip Technology INC.* [online]. 2010. [cit. 19.5.2011] Dostupné z WWW: <<http://www.microchip.com/stellent/idcplg>>
- [14] ZÁHLAVA, Vít. *Metodika návrhu plošných spojů*. skripta ČVUT. Praha 2000. 79 s.
- [15] MAXIM INTEGRATED PRODUCTS. *1-Wire Search Algorithm* 2002. 18. [online]. [cit. 19.5.2011] Dostupné z WWW: <<http://pdfserv.maximic.com/en/an/AN187.pdf>>
- [16] MICROCHIP TECHNOLOGY INC. *The Microchip TCP/IP Stack* 2002. 94. [online]. [cit. 19.5.2011] Dostupné z WWW: <<http://ww1.microchip.com/downloads/en/AppNotes/00833b.pdf>>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

RS232 Recommended Standard 232

USB Universal Serial Bus

TCP/IP Transmission Control Protocol/Internet Protocol

SMTP Simple Mail Transfer Protocol

RAM Random Access Memory

USART Universal Synchronous/Asynchronous Receiver/Transmitter

SPI Serial Peripheral Interface

I2C Inter Integrated Circuit

ADC Analog to Digital Converter

PWM Pulse Width Modulation

IEEE Institute of Electrical and Electronics Engineers

DMA Direct Memory Access

ISO/OSI International Organization for Standardization/Open Systems Interconnection

CSMA/CD Carrier Sense Multiple Access/Collision Detection

CRC Cyclic Redundancy Check

MAC Media Access Control

SCL Synchronous Clock

SDA Synchronous Data

ROM Read Only Memory

LSB Least Significant Bit

MSB Most Significant Bit

E2PROM Electrically Erasable Programmable Read Only Memory

MSSP Master Synchronous Serial Port

PLL Phase Lock Loop

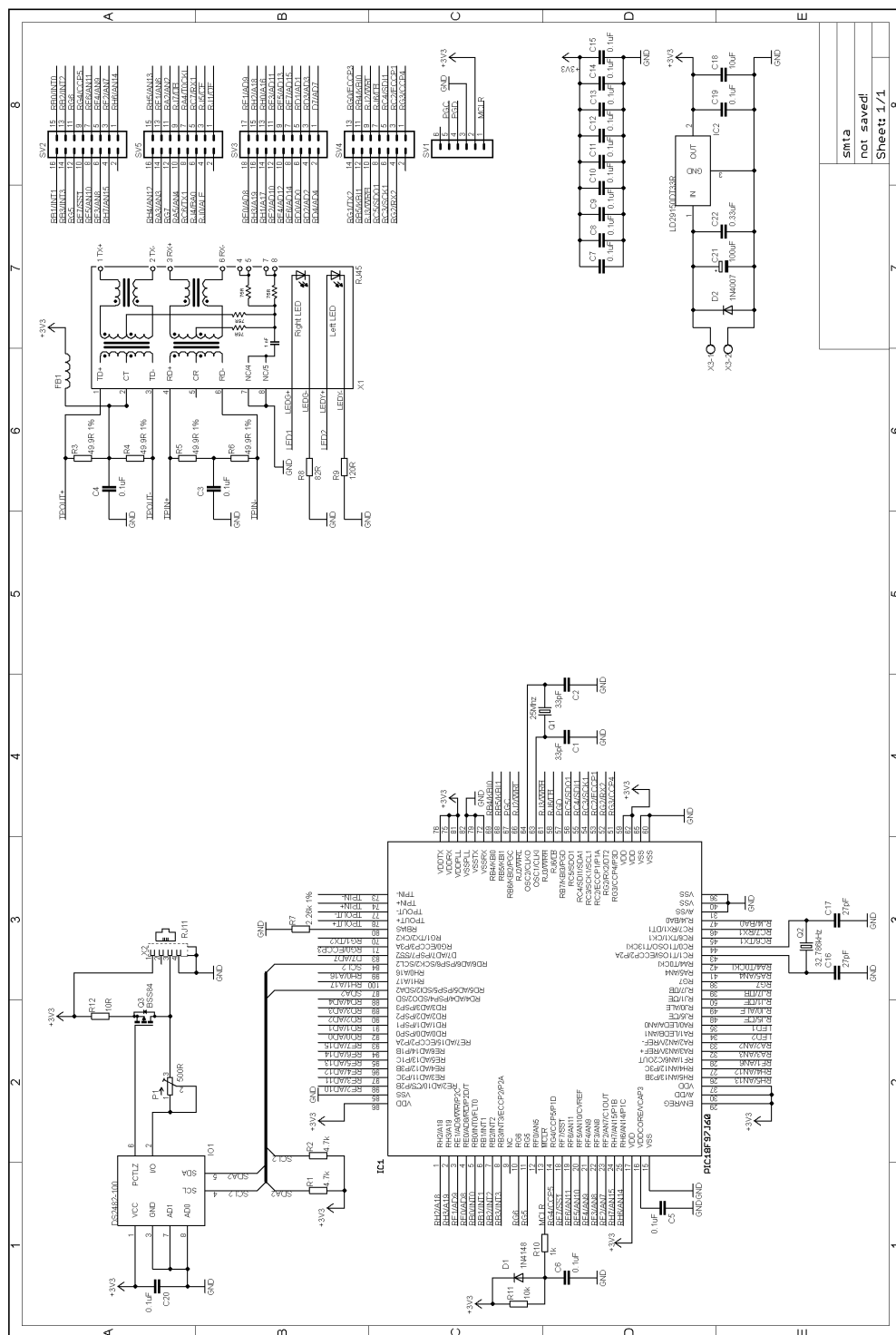
SSL Secure Socket Layer

SEZNAM PŘÍLOH

A	Výrobní dokumentace	64
A.1	Schema zapojení	64
A.2	Motivy plošných spojů	65
A.3	Osazovací výkresy	66
A.4	Motiv padů	67
A.5	Seznam součástek	68
A.6	Fotografie výrobku	70

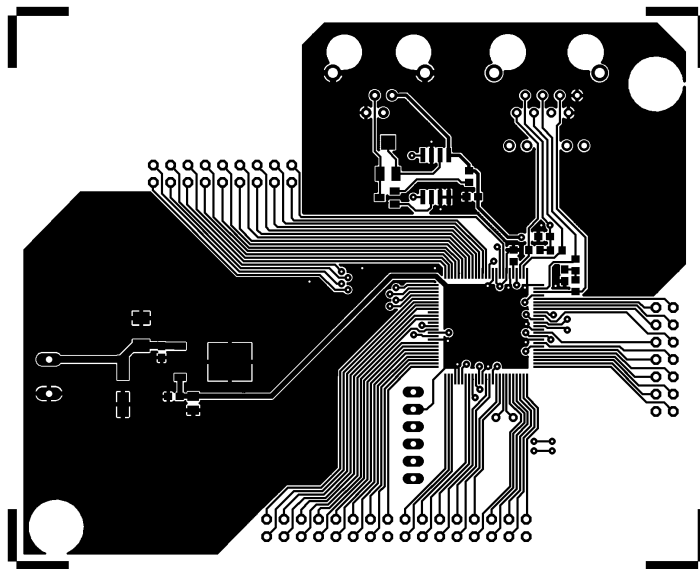
A VÝROBNÍ DOKUMENTACE

A.1 Schema zapojení

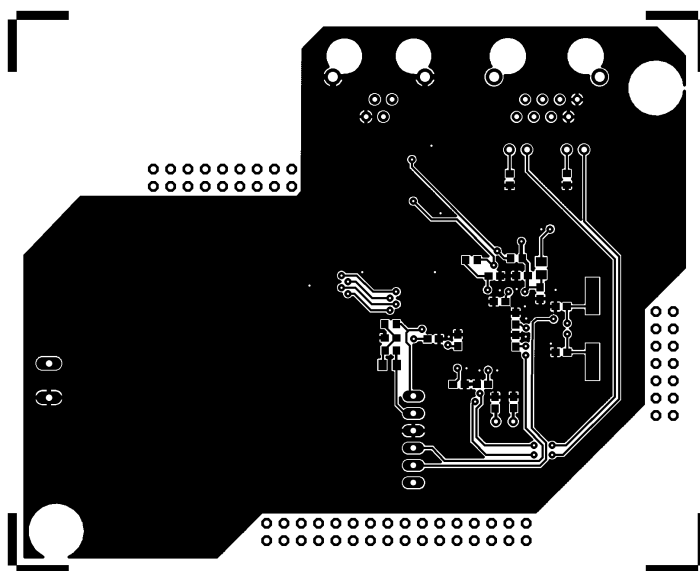


Obr. A.1: Schema zapojení

A.2 Motivy plošných spojů

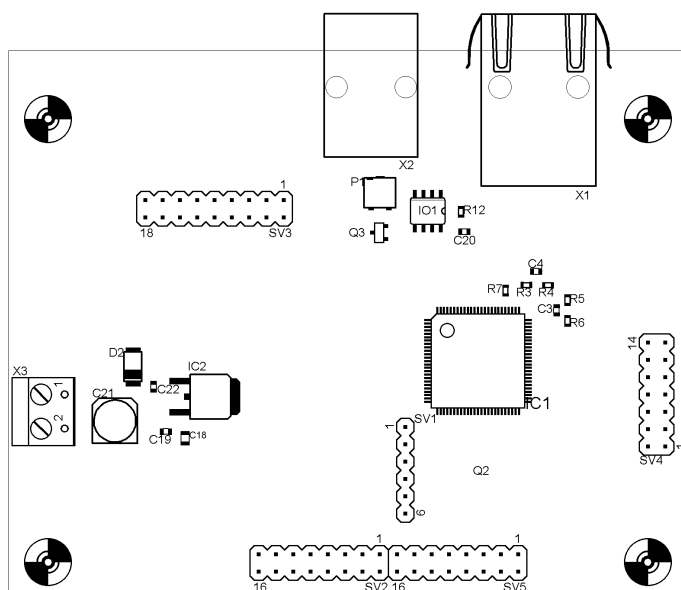


Obr. A.2: Motiv spojů horní strany desky

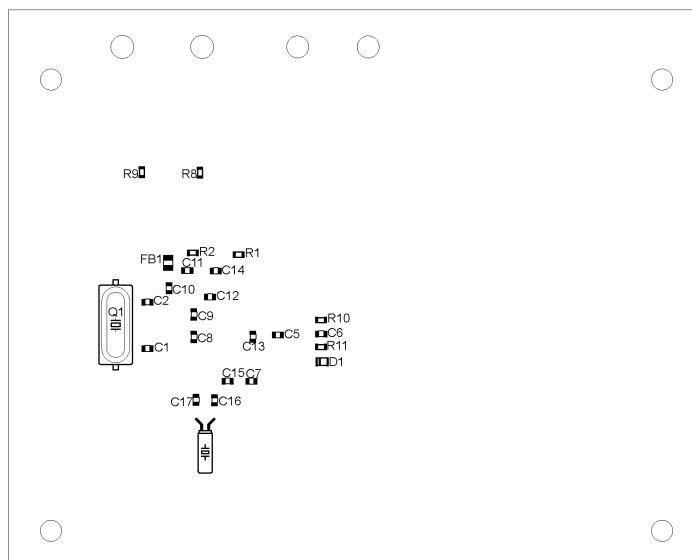


Obr. A.3: Motiv spojů spodní strany desky

A.3 Osazovací výkresy

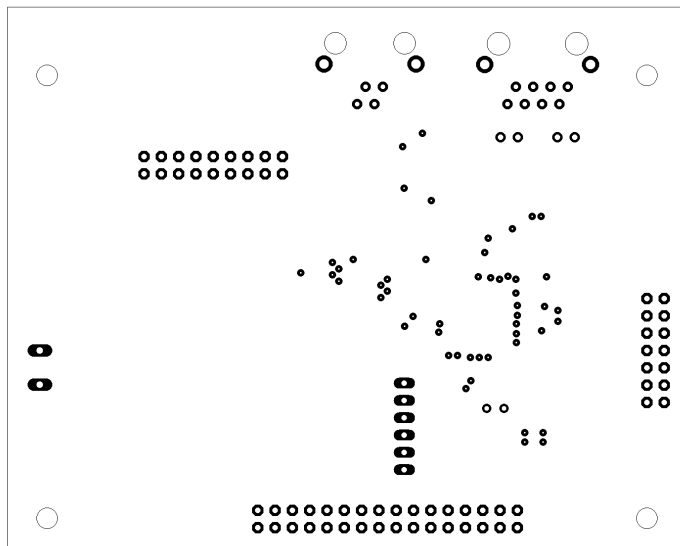


Obr. A.4: Osazovací výkres horní strany desky



Obr. A.5: Osazovací výkres spodní strany desky

A.4 Motiv padů



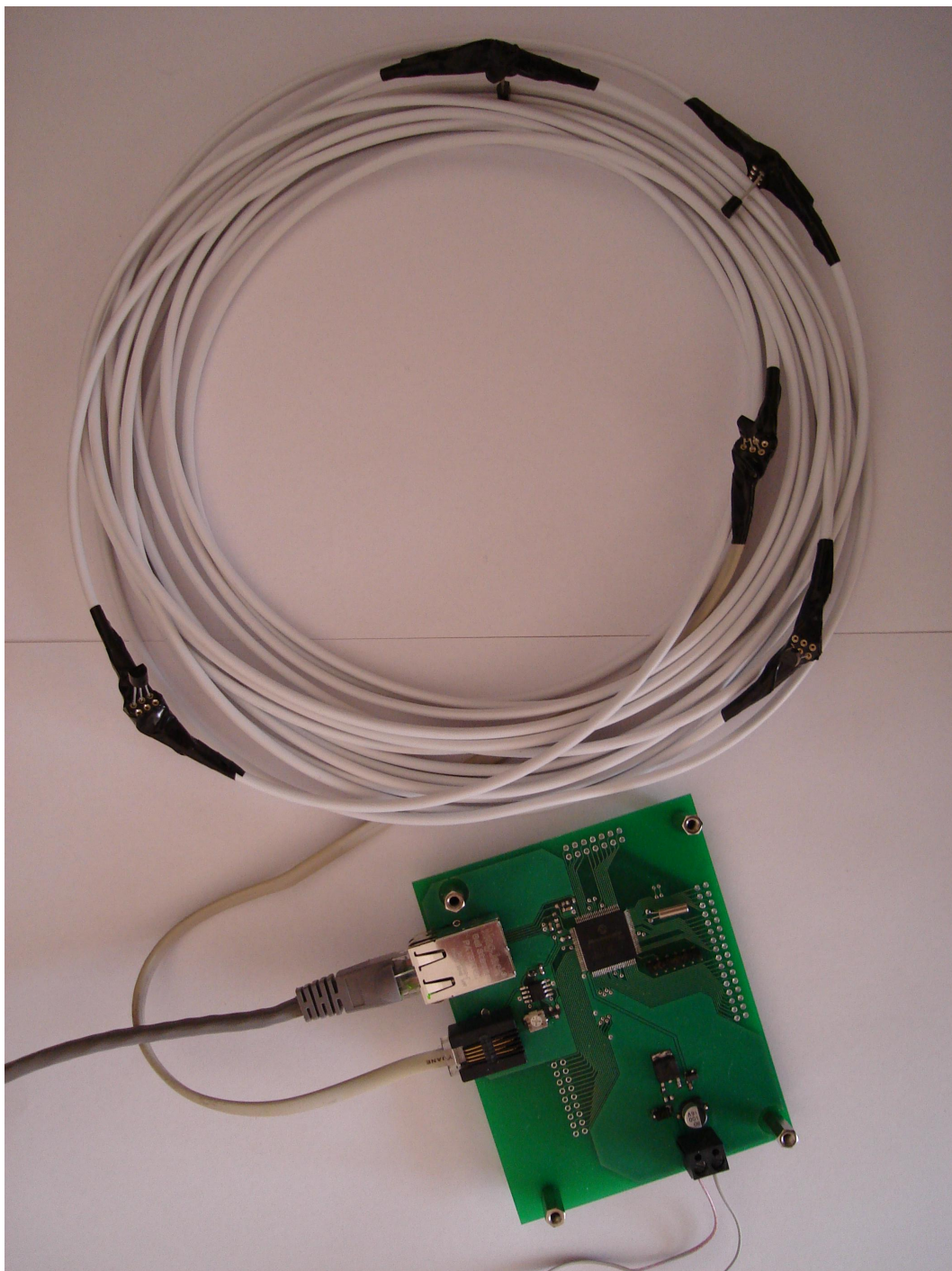
Obr. A.6: Motiv padů desky

A.5 Seznam součástek

Součástka	Hodnota	Pouzdro
C1	33pF	C0603
C2	33pF	C0603
C3	0.1uF	C0603
C4	0.1uF	C0603
C5	0.1uF	C0603
C6	0.1uF	C0603
C7	0.1uF	C0603
C8	0.1uF	C0603
C9	0.1uF	C0603
C10	0.1uF	C0603
C11	0.1uF	C0603
C12	0.1uF	C0603
C13	0.1uF	C0603
C14	0.1uF	C0603
C15	0.1uF	C0603
C16	27pF	C0603
C17	27pF	C0603
C18	10uF	C0805K
C19	0.1uF	C0603
C20	0.1uF	C0603
C21	100uF	153CLV-0605
C22	0.33uF	C0603
D1	1N4148	MICROMELF-W
D2	1N4007	DO-214AC
FB1		C0805
IC1	PIC18F97J60	TQFP100
IC2	LD29150DT33R	D-PAK_TO252AA
IO1	DS2482-100	SO08
P1	500R	SM-42/43A
Q1	25Mhz	HC49UP
Q2	32.786kHz	TC26H
Q3	BSS84	SOT23
R1	4.7k	R0603
R2	4.7k	R0603

Součástka	Hodnota	Pouzdro
R3	49.9R 1%	R0603
R4	49.9R 1%	R0603
R5	49.9R 1%	R0603
R6	49.9R 1%	R0603
R7	2.26k 1%	R0603
R8	80R	R0603
R9	120R	R0603
R10	1k	R0603
R11	10k	R0603
R12	10R	R0603
SV1	-	MA06-1
SV2	-	MA08-2
SV3	-	MA09-2
SV4	-	MA07-2
SV5	-	MA08-2
X1	RJ45	J1
X2	RJ11	XRJK-S-01-6-4-0-9M2
X3	-	W237-102

A.6 Fotografie výrobku



Obr. A.7: Výsledný výrobek