ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ v Praze

Fakulta elektrotechnická
Katedra řídicí techniky

Komunikační sběrnice LON (Local Operating Network)

Diplomová práce

2004 Martin LINHART

ČVUT Praha

Abstrakt

Komunikační sběrnice LON (Local Operating Network) je určena pro aplikace, kde je upřednostňována rozsáhlost sítě před rychlostí komunikace. Její výhodou je možnost zabezpečeného přístupu ke kritickým službám. Komunikačním protokolem je LonTalk. Je to kolizní „Predictive CSMA“ protokol. Základ každého zařízení operujícího v síti LON je procesor Neuron Chip.

Teoretická část této diplomové práce poskytuje potřebné informace o funkci protokolu LonTalk, rozebírá vlastnosti procesorů Neuron Chip a ve stručnosti shrnuje základy programovacího jazyka Neuron C. Ve druhé části je uveden návrh zařízení komunikujícího po sběrnici LON, včetně aplikace pro Neuron Chip.

Abstract

Communication bus LON (Local Operating Network) is intended for applications, where extensiveness of network to communication speed is prefered. Its benefit is secured access to protected services. Communication protocol is named LonTalk. It is a collisional "Predictive CSMA" protocol. Base of each device operating in the LON network is a Neuron Chip processor.

Theoretical part of this thesis provides necessary information about function of the LonTalk protocol, analyses features of the Neuron Chip processors and briefly summarizes the base of Neuron C programming language. In second part of this document is specified detailed design of the device which communicates over the LON, including the application.

Prohlášení

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

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejcích s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne: ..... Podpis: .....

Poděkování

Chtěl bych poděkovat všem, kdo mi přímo či nepřímo pomáhali při vzniku této práce. Zvláště pak děkuji Ing. Ondřejovi Dolejšovi, vedoucímu mé diplomové práce a Ing. Přemyslu Šůchovi, který mi vždy ochotně radil zvláště při oživování hardwarové části. Dále pak Ing. Jiřímu Goldmanovi za nasměrování při přepínání komunikačního módu procesoru a Ing. Dominiku Petráňovi za věnované komunikační rozhraní a cenné rady.

Velký dík také patří rodičům a přítelkyni, protože mi v průběhu studia poskytovali zázemí a veškerou podporu hmotnou i duševní.

Obsah

1 Úvod **3**

2 Technologie LonWorks **4**

3 Specifikace protokolu LonTalk **5**
3.1 Názvy a adresování 5
3.1.1 Význam jednotlivých položek v adrese 6
3.1.2 Formáty adres, použití 7
3.2 Model ISO/OSI 8
3.2.1 Fyzická vrstva 9
3.2.2 Spojovací vrstva 9
3.2.3 Síťová vrstva 10
3.2.4 Transportní vrstva 11
3.2.5 Relační vrstva 13
3.2.6 Prezentační a Aplikační vrstva 13
3.3 Mezivrstva pro přístup na médium 14
3.3.1 Určení okamžiku náhodného přístupu k médiu 14
3.3.2 Minimální hodnoty Beta1 a Beta2 15
3.3.3 Backlog 15
3.3.4 Priorita 15
3.3.5 Časování 16

4 Neuron Chip **17**
4.1 Paměť 19
4.2 Vstupy a Výstupy 20
4.3 Režimy komunikace 21
4.3.1 Single–Ended mód 22
4.3.2 Diferenční mód 23
4.3.3 Special–Purpose mód 24

5 Programovací jazyk Neuron C **25**
5.1 Odlišnosti od ANSI C 25
5.2 Nové datové typy a direktivy kompilátoru 26
5.2.1 Třída proměnných „network“ 26
5.2.2 Direktivy kompilátoru 27
5.2.3 Časovače a typy I/O objektů 28
5.2.4 SNVT proměnné 29
5.3 Vývoj aplikace 30

6 Pyroelektrický senzor infračerveného záření **32**

7 Realizace zařízení **33**
7.1 Fyzická část - Hardware 34
7.1.1 Analogová část - IR čidlo pohybu 34
7.1.2 Procesorová část 37

7.1.3 Přípravek SMX-DC1250 38
7.2 Aplikace pro NeuronChip – Firmware 39
7.2.1 Standardní profily 39
7.2.2 Obsluha událostí 42
7.3 Přepnutí režimu komunikace 46

8 Stručný přehled vývoje a integrace zařízení do sítě 48

9 Závěr 51

A Elektrická zapojení 55
A.1 Analogová část 55
A.2 Procesorová část 56
A.3 SMX-DC1250 57

B Seznamy součástek 58
B.1 Analogová část 58
B.2 Procesorová část 59
B.3 SMX-DC1250 60

C Přehled užitých zkratek a pojmů 61

D Obsah přiloženého CD 63

1 Úvod

Ve světě, a v posledních několika letech i v České republice, roste poptávka po možnostech automatizace budov. Jedná se zejména o velká obchodní centra a kancelářské prostory, ale i výrobní závody. Ve všech výše uvedených oblastech rostou nároky na bezpečnost, komfort a v neposlední řadě také na variabilitu užívaných prostor. Z finančního a časového hlediska je velice náročné provádět úpravy v elektrické instalaci. V případě, že jsou prvky propojeny pouze virtuálně (na úrovni síťového protokolu) je případná změna „zapojení“ dílem několika okamžiků.

V minulých letech si výrobci elektroinstalační a zabezpečovací techniky specifikovali vlastní uzavřené systémy komunikace mezi jednotlivými čidly a akčními prvky. Tento přístup omezuje výběr na zařízení od jediného výrobce, protože systémy jsou často navzájem neslučitelné. Právě tato vzájemná nekompatibilita vede v okamžiku, kdy zákazník požaduje část technologie od jiného výrobce, k drahým a komplikovaným řešením.

Společnost Echelon si tuto skutečnost uvědomila a svou sběrnici LON (Local Operating Network) uvolnila dalším výrobcům. Zároveň vydala komerční verze vývojových prostředí NodeBuilder a LonBuilder. Výhodou této technologie je její otevřenost a možnost komunikace mezi zařízeními od různých výrobců.

Tato práce je rozdělena na dvě části. V první části jsou vytvořeny teoretické základy, které jsou důležité pro seznámení se s daným tématem. V kapitole 3 je popsán komunikační protokol LonTalk, jenž je základem technologie LonWorks. Tato technologie využívá speciální procesory Neuron Chip, které jsou navrženy pro snadný vývoj uživatelských aplikací a zároveň mají implementován síťový protokol LonTalk. Procesorům Neuron Chip je věnována kapitola 4. Kapitola 5 obsahuje stručný popis programovacího jazyka Neuron C, který vychází z ANSI C, ale obsahuje určité odlišnosti.

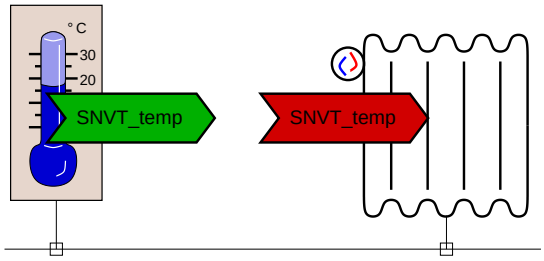
V druhé části je popsán vývoj zařízení (infračerveného čidla pohybu), které je vystavěno na procesoru Neuron Chip. Je tak schopné komunikovat v síti LON s ostatními zařízeními (poplachová ústředna, spínač osvětlení, atd.). Kapitola 6 obsahuje popis pyroelektrického senzoru, v kapitole 7, která je rozčleněna do dvou částí, je pak navrženo samotné zařízení. První část obsahuje návrh hardware, druhá pak vývoj aplikace pro Neuron Chip.

Práce je doplněna přílohami se schematy jednotlivých částí, seznamem součástek a slovníčkem užitých termínů. Přiloženo je také CD s doplňkovými informacemi a katalogovými listy použitých součástek.

2 Technologie LonWorks

Jak již bylo řečeno v úvodu, při stavbě budov se stále více uplatňují inteligentní prvky propojené pomocí sítě do distribuované technologie. LonWorks poskytuje v tomto směru otevřené řešení pro velké množství výrobců a tím i možné konkurenční prostředí.

Komunikačním protokolem technologie LonWorks je protokol LonTalk. Principiálně by se dal přirovnat k Ethernetu rozšířenému o nové možnosti. Jde o kolizní protokol, ale díky prediktivním vlastnostem radikálně zvyšuje průchodnost sítě při vysokém vytížení.



Obrázek 2.1: SNVT proměnné vytvářejí v podstatě virtuální spojení dvou členů síťě.

Podstatným přínosem jsou standardizované prostředky pro předávání a sdílení dat mezi jednotlivými částmi distribuované technologie. Jsou jimi tzv. SNVT proměnné (Standard Network Varible Type). Typ SNVT objektu jednoznačně určuje druh přenášené veličiny včetně rozsahu a fyzikálních jednotek. Každé zařízení má potom z pohledu ze sítě vstupní a výstupní síťové proměnné, s jejichž pomocí je definováno jeho síťové rozhraní. Z pohledu uživatele je tato skutečnost skryta a je vytvořeno „virtuální“ spojení, jak je vidět na obrázku 2.1. To je podstatné zjednodušení, které klade spolu s volnou topologií sítě menší nároky na pracovníky, kteří budou technologii instalovat.

Další vlastnosti systému LonWorks zmíněné jen v bodech:

- nové nastavení síťových proměnných je provedeno bez nutnosti zasahovat do kódu aplikace
- uživatelskou aplikaci je možné nahrát po síti LonWorks
- Pokud jsou použita komunikační rozhraní FTT-10A, je možné vést topologii sítě naprosto volně

3 Specifikace protokolu LonTalk

Protokol LonTalk byl navržen pro komunikaci mezi „inteligentními“ senzory, ovládacími a akčními prvky a dalšími členy v průmyslové síti. Tento typ sítí je charakterizován malými objemy přenášených dat, nízkými náklady na jednotlivé uzly (nódy), velkým množstvím rozličných komunikačních médií a často náročnými pracovními podmínkami (elektromagnetické rušení, prach…).

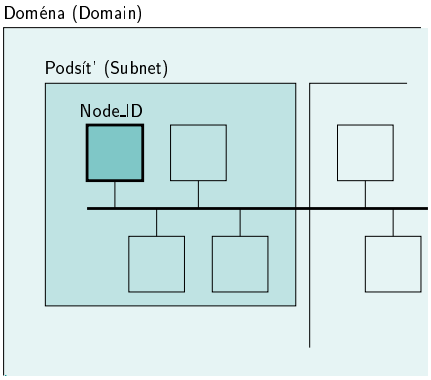
V kapitole 3.1 jsou shrnuty způsoby adresování cílových nódů. Popisy jednotlivých vrstev protokolu LonTalk vztažených k modelu ISO/OSI jsou uvedeny v kapitole 3.2.

LonTalk poskytuje prediktivní přístup ke komunikačnímu médiu. O to se stará mezivrstva pro přístup na médium, která je podrobněji popsána v kapitole 3.3.

3.1 Názvy a adresování

LonTalk používá hierarchické členění adres. Pro tři základní typy vypadá řazení jednotlivých částí následovně:
Domain/Subnet/Node
Domain/Subnet/Neuron_ID
Domain/Group/Member

Každá PDU[1] obsahuje výchozí i cílovou adresu. Adresování obstarávají společně vrstvy 3 a 4 modelu ISO/OSI (viz. obrázek 3.4). Části Domain a Subnet v adrese slouží pro směrování, které zajišťuje síťová vrstva. Pole Node označuje konkrétní nód. Používá se místo Neuron_ID z důvodu snadné výměny (např. při poruše původního zařízení). Pole Node se v těchto případech nemění, jen je nahrána konfigurace do nového nódu. Naopak Neuron_ID má každý Neuron Chip unikátní, přidělené při výrobě. Zůstává tedy stejné i při přesunutí do jiné části sítě.



Obrázek 3.1: Významy jednotlivých částí síťové adresy nódu.

Délka celé adresy je proměnná. Pohybuje se v rozmezí 0, 1, 3 nebo 6 bytů.

[1] PDU – Protocol Data Unit

Velikost pole subnet je 8 bitů, Node má délku 7 bitů. Jak je vidět na obrázku 3.1, tyto dvě části určují maximální počet nódů v jedné doméně, tedy $2^7 \cdot 2^8 \approx 2^{15}$ (Adresy, které obsahují samé nuly se nepoužívají). Pole Group je dlouhé 6 bitů, Neuron_ID pak 48 bitů.

3.1.1 Význam jednotlivých položek v adrese

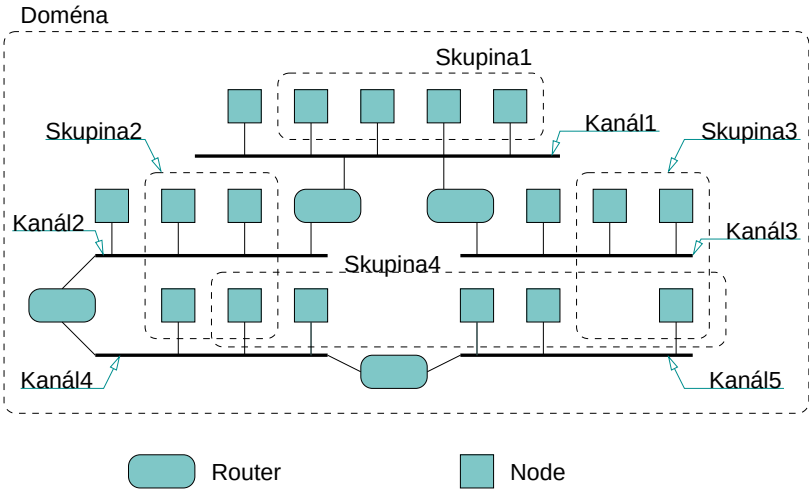
Na obrázku 3.3 jsou uvedeny různé způsoby adresování cílových nódů. Jednotlivá pole mají následující význam:

Neuron_ID – Toto označení se používá pouze při instalaci nódu. Poté je mu přidělena v síti jedinečná kombinace Domain/Subnet/Node, popř. je také zařazen do jedné či více skupin. Neuron_ID je 48-bitové číslo, unikátní pro každý NeuronChip. Je procesoru přiděleno během výroby.

Domény (Domain) – Doména je největší celek, ve kterém probíhá komunikace na úrovni LonTalk protokolu. Je to v podstatě virtuální síť, na kterou je omezena veškerá komunikace. Kde je nutná komunikace mezi doménami, je nutné zapojit gateway, která pracuje na úrovni aplikační vrstvy.

Podsíť (Subnet) – Po adrese domény následuje v adresní hierarchii adresa podsítě. Tato určuje jednoznačně podsíť v dané doméně. Pokud je hodnota 0, znamená to, že podsíť není definována nebo je neznámá. V jedné podsíti může být připojeno 0-127 nódů. Jedná se v podstatě o virtuální kanál, nezávislý na fyzickém propojení. To znamená, že na jednom fyzickém vedení může být několik podsítí, ale také jednu podsíť může tvořit několik odlišných přenosových médií navzájem propojených pomocí síťových prvků typu bridge nebo repeater.

Pokud je subnet 0, znamená to, že nód nezná adresu své domény.



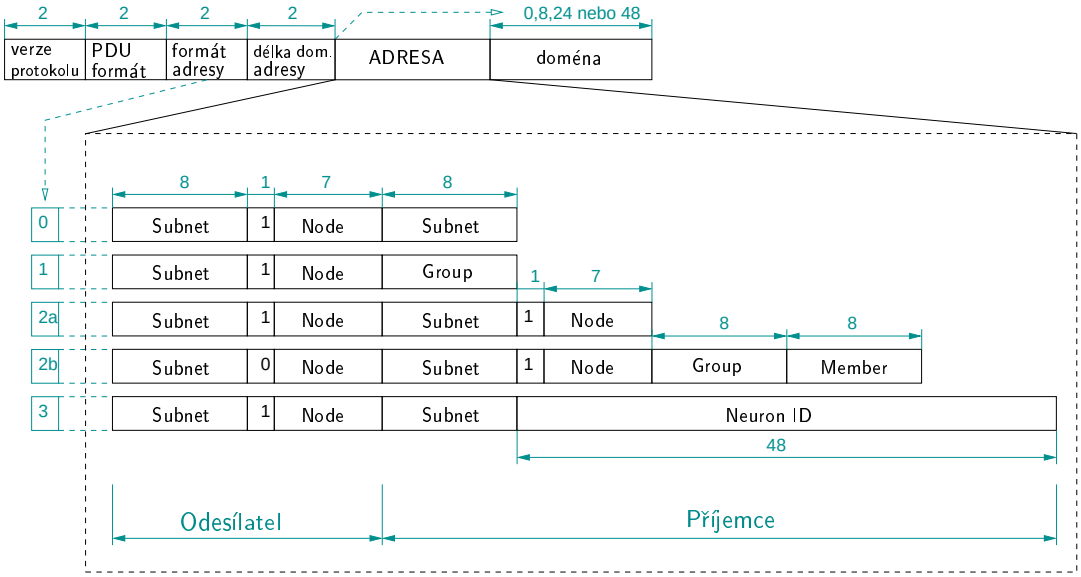
Obrázek 3.2: Příklady fyzického a logického uspořádání uzlů v doméně.

Nód (Node) – Označuje konkrétní nód v jedné podsíti. Rozdíl mezi Node a Neuron_ID by se dal přirovnat k rozdílu mezi MAC a IP adresou síťové karty počítače v síti internet. Při případné výměně nódu (např. při poruše) za nový je pouze u nového nahrána síťová konfigurace. V případě, že by se používalo pro běžnou komunikaci Neuron_ID, musely by se změnit konfigurace všech nódů, které s vyměněným kusem komunikovaly. Fyzicky může být nód připojen do dvou podsítí. Pokud tomu tak je, je pro každou podsíť vytvořen vlastní identifikátor nódu.

Skupina (Group) – Číslo skupiny stojí na druhém místě v adresní hierarchii LonTalk protokolu. Adresování po skupinách usnadňuje komunikaci typu multicast[2].

3.1.2 Formáty adres, použití

Na obrázku 3.3 je znázorněna skladba adresní části hlavičky paketu. Část, ve které je uložena výchozí adresa, má pevnou strukturu, cílová adresa má několik formátů.



Obrázek 3.3: Fyzické formáty adresy.

Formát 0 Slouží k zasílání zpráv typu broadcast[3]. Pokud je cílová podsíť 0, zašle se zpráva do všech podsítí. Číslo 1–255 pak určuje konkrétní podsíť.

Formát 1 Slouží k zasílání zpráv typu multicast. Jako cílová adresa slouží číslo skupiny, které má být zpráva doručena.

Formát 2 Má dvě varianty:

[2]Multicast – jedna zpráva je současně zaslána více účastníkům.
[3]Broadcast - zpráva je zaslána všem účastníkům v síti

2a. slouží k doručování unicast[4] zpráv a potvrzení[5].

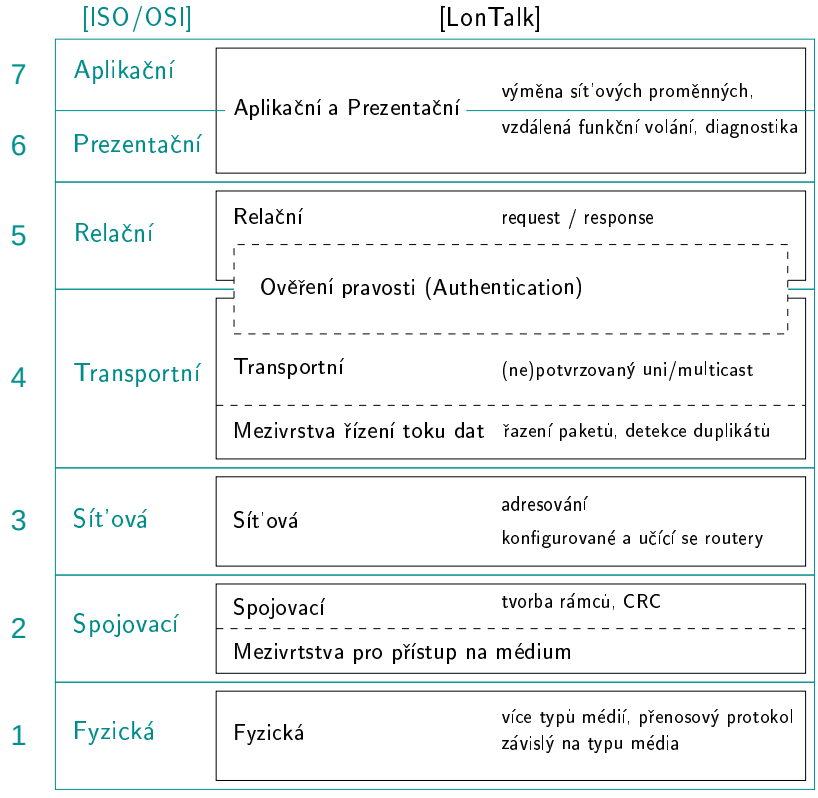
2b. slouží k doručování potvrzení pro skupiny.

Formát 3 Slouží k adresování nódu pomocí jeho unikátního Neuron_ID. Toto číslo smí být použito jen jako cílová adresa. Neuron_ID je jedinečné 48 bitů dlouhé číslo, které je Neuron Chipu přiděleno při výrobě.

Pokud není nód nakonfigurovaný, přijímá zprávy zaslané pro všechny domény a podsítě. Z těch vybírá jen ty, co jsou zaslané na jeho Neuron_ID. Po zadání parametrů může jeden nód příslušet do jedné nebo dvou domén.

3.2 Model ISO/OSI

Protokol LonTalk může být popsán pomocí sedmi vrstev ISO/OSI[6] modelu, jak je vidět na obrázku 3.4.



Obrázek 3.4: Vrstvy LonTalk protokolu v ISO/OSI modelu.

[4]Unicast – přímá komunikace mezi dvěma účastníky. Jedna zpráva je zaslána právě jednomu příjemci.

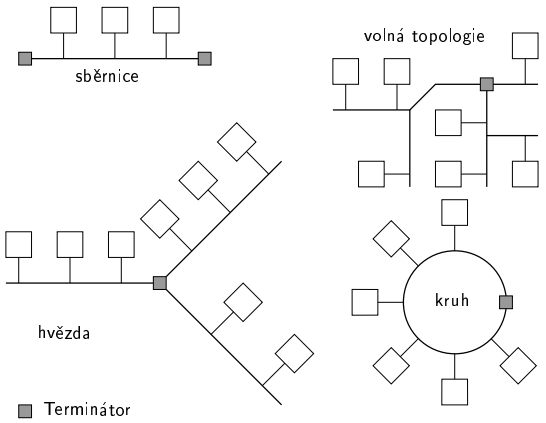
[5]Potvrzení – pokud je o to uzel požádán, zašle po úspěšném přijetí paketu potvrzení o přijetí.

[6]ISO/OSI (International Standard Organization/Open System Interconnection)

3.2.1 Fyzická vrstva

Fyzická vrstva definuje typ média, po kterém se bude komunikovat (koaxiální kabel, kroucená dvoulinka, optické vedení a další). Od toho se pak odvozuje způsob, jakým se budou kódovat jednotlivé bity, popř. byty. Jedna z možností, kterou používá technologie LonWorks, je diferenční manchester. Princip je vysvětlen v kapitole 4.3.

LonWorks nemá fyzickou vrstvu pevně specifikovanou. Existují komunikační rozhraní (transceivery) pro několik typů přenosových kanálů. Rozšířenými typy jsou PowerLine nebo FTT-10A. Druhý zmíněný má výhodu tzv. volné topologie. Při instalaci technologie není potom nutné omezovat se svazujícími pravidly topologie. Příklady topologií jsou na obrázku 3.5.



Obrázek 3.5: Topologie sítí.

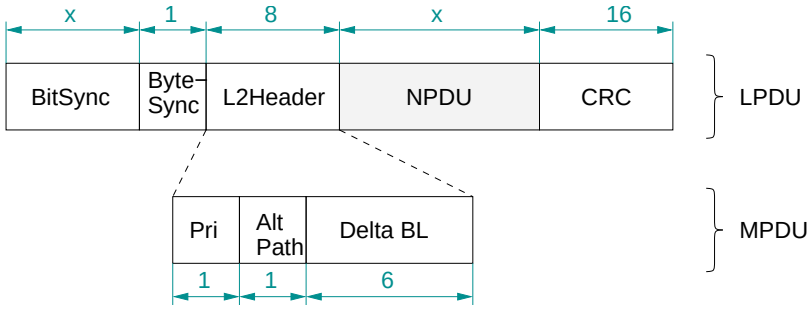
3.2.2 Spojovací vrstva

Tato vrstva poskytuje službu doručování seřazených a nepotvrzovaných paketů MPDU[7]/LPDU[8] v rámci jedné podsítě. Odchozí data jsou zde dělena do rámců a předávána fyzické vrstvě k odeslání. U přijatých paketů jsou pomocí CRC[9] algoritmu detekovány případné chyby. Ty však nejsou opravovány a chybný paket je odstraněn. Protože většina komunikačních rozhraní nemá možnost detekce kolizí, jsou pomocí tohoto algoritmu rozeznávány také kolize na lince. Na obrázku 3.6 je znázorněn tvar paketu MPDU/LPDU. Tento tvar koresponduje s obrázkem 4.3, na kterém jsou patrné významy synchronizačních polí `BitSync` a `ByteSync`. Popis pole `L2Header` je uveden v kapitole 3.3.

Mezivrstva pro přístup na médium

Pomocí mezivrstvy pro přístup na médium (MAC[10]) poskytuje spojovací

[7]MPDU (Media access control layer Protocol Data Unit)
[8]LPDU (Link layer Protocol Data Unit)
[9]CRC (Cyclic Redundancy Check) ... způsob detekce chyb pomocí kontrolních součtů
[10]MAC (Media Access Control) ... řízení přístupu k fyzickému médiu



Obrázek 3.6: Formát kombinovaného rámce MPDU/LPDU.

vrstva také přístup k přenosovému médiu. Protože se jedná o složitější proces, je mu věnována kapitola 3.3.

3.2.3 Síťová vrstva

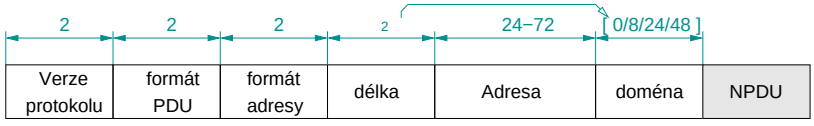
Zde je vybírán způsob adresování odchozích zpráv. LonTalk rozeznává několik typů cílových adres a tím i způsobů vysílání:

- Unicast – Zpráva je odeslána přímo jednomu příjemci.
- Multicast – Zpráva je odeslána skupině příjemců (Group). Příkladem může být odeslání povelu žaluziím na jedné straně domu.
- Broadcast – Zpráva je odeslána všem nódům v doméně.

Parametry, které určují, komu bude který nód zasílat svá data se nazývají síťová nastavení, v terminologii LonWorks bindings. Tyto parametry jsou zadány až v okamiku instalace síťě příslušným softwarovým nástrojem (Network Management Software).

Síťová adresa každého nódu se skládá ze tří částí: domény/podsítě/node_ID. Významy uvedených částí adres jsou znázorněny na obrázku 3.1 a dají se připodobnit k poštovní adrese město/ulice/dům.

Pro snazší adresování většího počtu nódů najednou se používá dělení do skupin (Groups). Zpráva tak může být rozeslána celé skupině nódů najednou. Nódy příslušející do jedné skupiny nemusí být pouze z jedné podsítě, jak je vidět na obrázku 3.3. Toto členění může mít uplatnění např. při ovládání všech světel v jedné místnosti.



Obrázek 3.7: Formát NPDU.

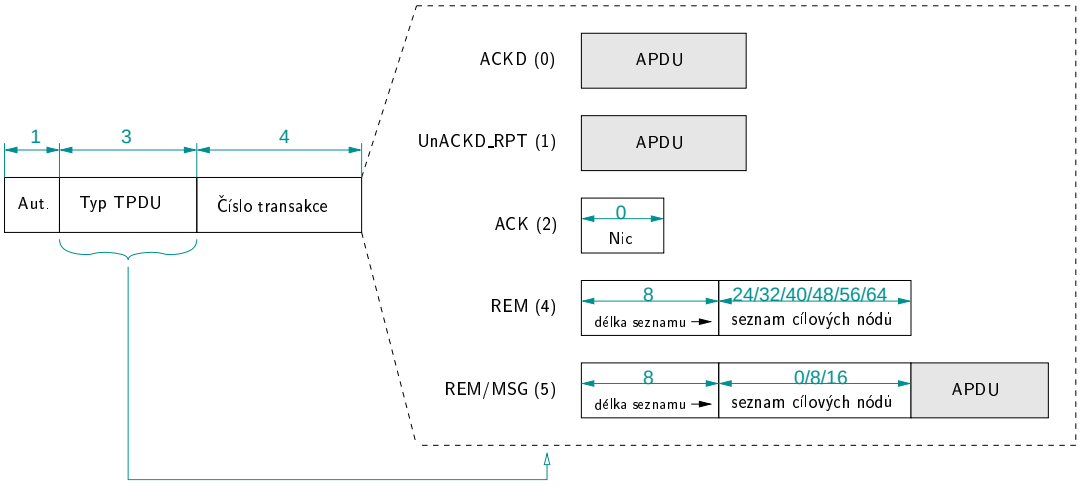
Na úrovni síťové vrstvy pracují v LonWorks směrovače (Router). Ty existují ve dvou verzích. Samoučící se a ručně konfigurované. Pokud je použito komunikační rozhraní, jež umožňuje volnou topologii sítě (např. FTT-10A), mohou na

vedení vzniknout smyčky. V takových případech hrozí zacyklení paketu. Z tohoto důvodu musí být v sítích, kde uvedené smyčky exisují, použity ručně konfigurované routery. S jejich pomocí je vytvořen logický strom bez cyklů.

Formát NPDU[11] je na obrázku 3.7. Čísla nad každým políčkem vyjadřuje délku v bitech.

3.2.4 Transportní vrstva

Transportní vrstva zajišťuje ověření, zda přenos zprávy proběhl úspěšně – tzv. Acknowledgments. Pokud je zpráva odeslána s žádostí o potvrzení („acknowledged message“), očekává vysílající nód od příjemce potvrzení o přijetí. Jestliže jej do určité doby nedostane, odešle nódu tzv. reminder message. Transportní vrstva vytváří několik typů TPDU[12], jak je vidět na obrázku 3.8. V rámečcích



Obrázek 3.8: Formát TPDU.

jsou naznačeny významy jednotlivých částí, nad rámečky jsou uvedeny délky v bitech. Zkratka Aut. v prvním políčku TPDU na obrázku 3.8 vyjadřuje požadavek k ověření Authentication pomocí 48-mi bitového tajného klíče. Mechanismus je popsán níže v kapitole. Významy jednotlivých typů TPDU na obrázku 3.8:

- ACKD – Potvrzovaná zpráva se používá při zahájení komunikace. Vysílá se jen s formáty adres 1, 2a a omezeně s formátem 3 (viz. obr. 3.3). Tento typ musí být potvrzen všemi uzly, kterým byl zaslán. Pokud je paket zasílán skupině, nemusí být vysílající nód jejím členem. Maximální počet nódů, které mohou být takto adresovány, je omezen na 63.
- UnACKD_RPT – Nepotvrzovaná-opakovaná zpráva. Přijetí této zprávy příjemce nepotvrzuje. Jednoduchá nepotvrzovaná zpráva neobsahuje TPDU hlavičku a proto nemůže být kontrolována na duplicity. V tomto případě není kladeno žádné omezení na počet adresovaných nódů. Jediné omezení je maximální počet nódů v doméně.

[11]NPDU (Network layer Protocol Data Unit)
[12]TPDU (Transport layer Protocol Data Unit)

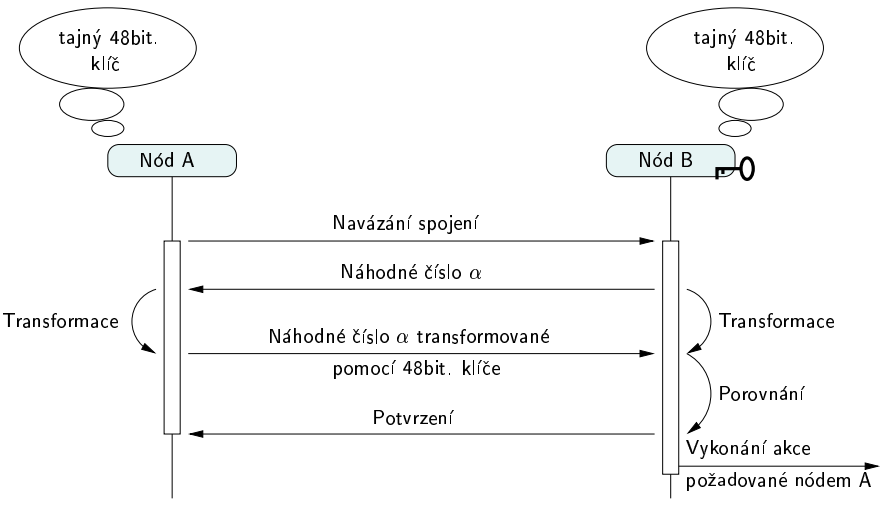
- ACK – Potvrzení přijetí. Využívá formát adresy 2a nebo 2b podle obrázku 3.3. Tělo této zprávy má nulovou velikost, číslo transakce je stejné jako při přijetí, což vyjadřuje přijetí daného paketu.
- REM/MSG – Opakovaná výzva/zpráva se používá v případě, kdy vysílající uzel nedostal odpověď od všech adresovaných uzlů ze skupiny, kde nejvyšší počet členů je 16. V tomto případě je opakovaně zaslána zpráva spolu s polem „Seznam cílových nódů“. Každá buňka tohoto pole obsahuje buď logickou jedna, která vyjadřuje, že vysílající uzel již potvrzení obdržel nebo logickou nulu, ta oproti předchozímu stavu vyjadřuje novou žádost o potvrzení.
- REM – Opakovaná výzva se používá v případech, kdy je členů více než 16. V tomto případě není na rozdíl od předchozího připojen APDU, zato pole se seznamem uzlů může být až 64bitů dlouhé.

Mezivrstva pro řízení toku dat

Protokol pro řízení toku dat používá 4-bitová čísla, která vyjadřují pořadová čísla transakcí. Vysílající nód jimi čísluje odchozí pakety, přijímající pak s jejich pomocí kontroluje správné řazení paketů. Předpokládá se, že síť tvoří strom, ve kterém se pakety navzájem nemohou předběhnout. V případě, že existuje více možných cest, předpokládá se, že všechny možné cesty zpozdí pakety přibližně stejně. Rozdíly v dobách šíření musí být menší, než nejkratší čas jedné potvrzované transakce. Jinými slovy nesmí dojít k přijetí paketů v jiném pořadí, než v tom, ve kterém byly vyslány. V případě, že se tak stane, jsou tyto zprávy vyhodnoceny jako duplicity.

Mezivrstva pro ověřování účastníka – Authentication

LonTalk umožňuje ověření pravosti nódu (Authentication), jenž navazuje spojení se zabezpečeným nódem. K tomu slouží 48-mi bitový klíč, jenž je uložen

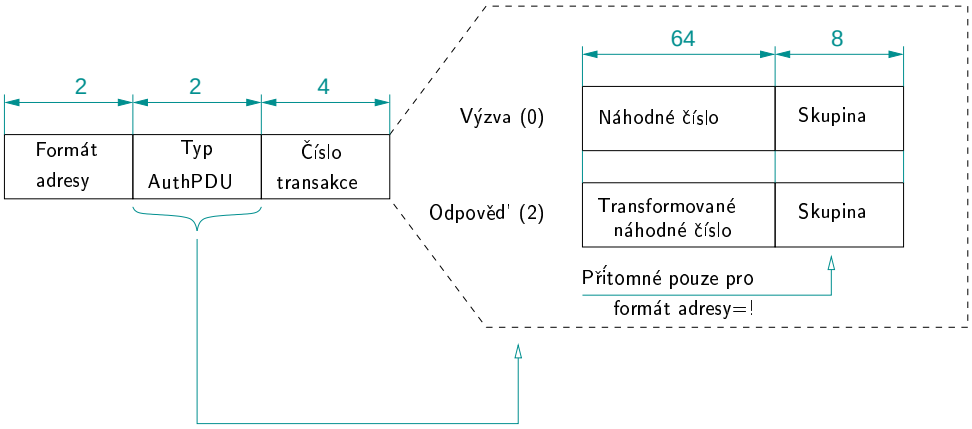


Obrázek 3.9: Princip mechanismu ověření účastníka pomocí 48bit. klíče.

do obou nódů při instalaci sítě. Mechanismus je patrný z obrázku 3.9. Tuto mezivrstvu využívají společně transportní a relační vrstva.

Nód, jenž je dotazován, zašle jako odpověď náhodné číslo, které zároveň transformuje pomocí svého tajného klíče. Nód, jenž relaci zahajoval, provede tu samou operaci se svým klíčem a výsledek zašle zpět. Pokud jsou obě transformace identické, je operace provedena. Formát AuthPDU[13] je uveden na obrázku 3.10.

Algoritmus, podle kterého je provedena transformace, je vlastnictvím společnosti Echelon. Ta jej z bezpečnostních důvodů nezveřejňuje.



Obrázek 3.10: Formáty AuthPDU.

3.2.5 Relační vrstva

Tato vrstva poskytuje tzv. Request–Response mechanismus. Tato služba umožňuje klientovi zaslat požadavek na vzdálený server a následně obdržet výsledek požadované operace. Pro tento typ relací zajišťuje ve spolupráci s mezivrstvou Authentication ověření nódů.

Na základě Request–Response mechanismu mohou být vystavěna vzdálená funkční volání typu Remote Procedure Call.

3.2.6 Prezentační a Aplikační vrstva

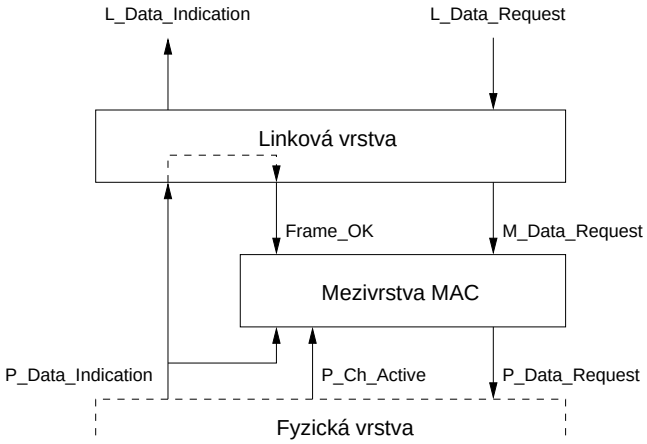
Prezentační a Aplikační vrstva tvoří základ pro vzájemné spolupůsobení nódů v síti, například pomocí tzv. SNVT[14] proměnných. Nódy tak mohou sdílet hodnoty bez znalosti dalších údajů o jejich poskytovateli, jako např. výrobce, typ zařízení a další.

[13] AuthPDU (Authentication sublayer Protocol Data Unit)

[14] SNVT (Standard Network Variable Type) – standardizovaný prostředek pro výměnu a sdílení dat mezi nódy v technologii LonWorks.

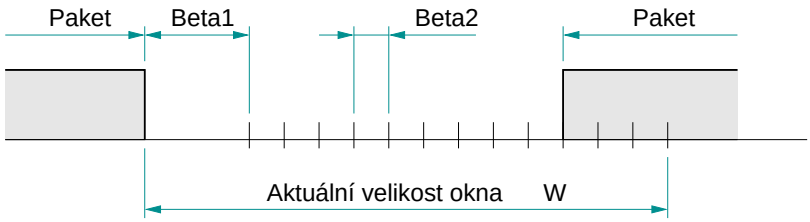
3.3 Mezivrstva pro přístup na médium

Komunikační protokol LonTalk pracuje na principu nedeterministického přístupu k přenosovému médiu. Tento přístup používá také například rozšířená počítačová síť Ethernet a označuje se CSMA/CD[15]. Pokud zařízení, připojené k síti Ethernet, potřebuje vyslat svá data, sleduje provoz na lince a vyčkává na její uvolnění. Jakmile nastane klid, náhodně vybere okamžik začáku vysílání z pevného intervalu. Při vysílání sleduje stav linky a pokud zaznamená kolizi, přeruší vysílání a celý proces se opakuje.



Obrázek 3.11: Rozhraní mezi Linkovou vrstvou a podvrstvou pro přístup na médium.

Protokol LonTalk pracuje na obdobném principu. Nód však při odposlechu na lince přijatá data také analyzuje a z jejich charakteru odhaduje, jak vytížená bude linka v následujících okamžicích. Podle tohoto odhadu potom upravuje šířku časového okna W, ze kterého vybírá náhodný okamžik pro začátek vysílání (Je-li očekáván velký provoz, bude okno široké a naopak). Tímto způsobem je snížena pravděpodobnost kolize a linka je tak více průchozí i při vyšších zatíženích. Tento přístup se označuje jako Predictive CSMA.



Obrázek 3.12: Prediktivní CSMA přístup na sběrnici LonTalk protokolu.

3.3.1 Určení okamžiku náhodného přístupu k médiu

Nód odposlouchává datový tok na sběrnici. Pokud nastane ticho, počká dobu vymezenou intervalem `Beta1` (viz. obr. 3.12). Jestliže je i na konci `Beta1` na

[15]CSMA/CD (Carrier Sense, Multiple Acces / Collision Detection)

lince klid, je médium uznáno za volné. Nód nyní vygeneruje náhodné číslo, které určí, kolik časových jednotek (timeslotů) `Beta2` počká, než začne vysílat. Pokud i na konci tohoto intervalu je sběrnice volná, dojde k vyslání paketu. Pokud se mezitím na lince objeví vysílání jiného nódu, situace se opakuje. Šířka okna se určuje podle vztahu

$$W = \mathrm{BL} \cdot w_{base} \tag{3.1}$$

kde

BL je tzv. *Back Log*, tedy koeficient, jehož velikost se určuje na základě odposlechnutých předchozích relací

w_{base} je základní šířka okna.

3.3.2 Minimální hodnoty Beta1 a Beta2

Pokud po dobu `Beta1` ani po jejím skončení není na lince zaznamenán žádný provoz, je vyhodnocena jako *volná*. Délka periody `Beta1` je určena podle vztahu:

$$\mathrm{Beta1} > db + (2 \cdot \tau_p + \tau_m) \tag{3.2}$$

kde

db je doba potřebná k přenesení jednoho bitu

τ_p vyjadřuje fyzikální závislosti na médiu (zpoždění závislé na délce média)

τ_m vyjadřuje dobu, která uplyne od okamžiku, kdy je zaznamenán klid na lince až do doby, kdy se může fyzicky objevit vysílaná informace na médiu. Jsou zde zahrnuta zpoždění uvnitř MAC mezivrstvy.

Node poté vyčká náhodně určený násobek timeslotu `Beta2`. Jeho minimální velikost je dána vztahem:

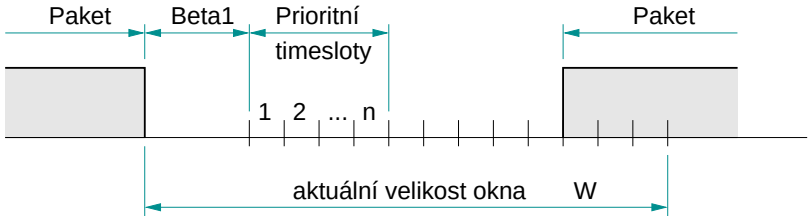
$$\mathrm{Beta2} > 2 \cdot \tau_p + \tau_m \tag{3.3}$$

3.3.3 Backlog

V kapitole 3.3.1 ve vztahu 3.1 je zmíněn koeficient BL. Na tomto koeficientu závisí správná funkce prediktivního CSMA. BL je inkrementován při příjmu nebo vyslání MPDU a dekrementován periodicky. Jeho hodnota nesmí klesnout pod 1. Dále, pokud je zaznamenána kolize při pokusu o přístup na médium, je BL zvýšen o jedna. Pokud kolize nastane 255-krát za sebou, je paket vyřazen.

3.3.4 Priorita

Protokol LonTalk rozlišuje z hlediska priority dva typy paketů: standardní a prioritní. V případě, že nód vysílá druhé jmenované, jsou do časového okna vřazeny tzv. prioritní timesloty. Z těch se vybírá náhodný čas, pokud je odchozí paket označen jako prioritní. Tato situace je naznačena na obrázku 3.13. Pokud je zaznamenána kolize při pokusu přenést zprávu s využitím prioritního slotu a to dvakrát po sobě, je další pokus vyslán jako paket bez priority.



Obrázek 3.13: Vřazení prioritních timeslotů do časového okna W.

3.3.5 Časování

Komunikační rychlost procesoru Neuron Chip je omezena jeho taktovací frekvencí, přímo ji také ovlivňuje použité komunikační rozhraní. Pohybuje se od 610bps do 2,5Mbps[16].

Komunikační port může být konfigurován v tzv. `Direct Connect` režimu. Neuron Chip potom řídí komunikační rychlost, kódování dat při vysílání a další. Druhou možností je `Special Purpose mode` (viz. kapitola 4.3.3), kdy zmíněné veličiny má na starosti přímo vysílač. V obou případech ale Neuron Chip obstarává hodnoty `Beta1` a `Beta2`. Vztah pro výpočet timeslotu `Beta2` je následující:

$$\text{Beta2} = CT \cdot (40 + 20 \cdot v) \tag{3.4}$$

Pro `Beta1` se výpočet liší podle konfigurace komunikačního portu.
V přímém režimu:

$$\text{Beta1} = CT \cdot (583 + \text{Beta2} + PAD) \tag{3.5}$$

Ve speciálním režimu potom:

$$\text{Beta1} = CT \cdot (577 + \text{Beta2} + PAD) \tag{3.6}$$

kde

CT je doba jednoho výpočetního cyklu Neuron Chipu.

v je dolaďovací koeficient, $v \in \langle 0;\ 255 \rangle$

PAD slouží ke kompenzaci komunikačních prodlev s nódy, které mají nižší taktovací kmitočet. PAD má opět dva způsoby výpočtu, lišící se podle velikosti v.

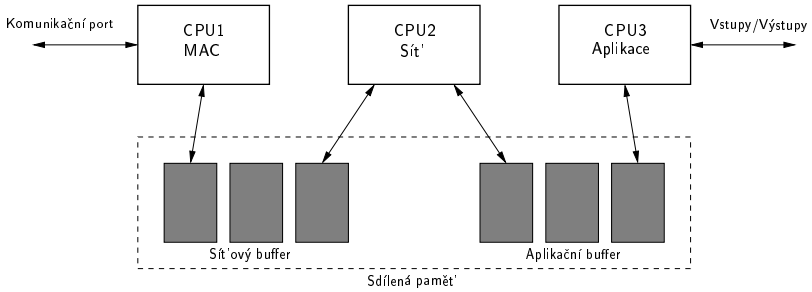
pro $v \in \langle 0; 128)$ $\ldots PAD = 41 \cdot v$

pro $v \in \langle 128; 256 \rangle$ $\ldots PAD = 145 \cdot (v - 128)$

[16]Pouze u procesorů, které podporují taktovací frekvenci 20MHz.

4 Neuron Chip

Procesor Neuron Chip je srdcem technologie LonWorks. Jedná se o mikroprocesor speciálně navržený pro rychlý a ekonomicky výhodný vývoj širokého spektra „inteligentních“ akčních prvků, senzorů, ale i nadřazených systémů. Skládá se ze tří 8-mi bitových samostatných procesorů, jak ukazuje obrázek 4.1. V současné době existují dvě základní řady: 3120xx a 3150xx, které se liší v uspořádání jednotlivých typů pamětí. Více je uvedeno v kapitole 4.1.



Obrázek 4.1: Uspořádání procesorů uvnitř TMPN3120xx/3150.

- CPU1 - První procesor se stará o řízení přístupu na médium - MAC[17], což zahrnuje částečně první a druhou vrstvu sedmivrstvého ISO/OSI modelu (viz. kapitola 3.2). S CPU2 komunikuje tento procesor pomocí Síťového bufferu, jak je vidět na obrázku 4.1.

- CPU2 - Je tzv. síťový procesor. Obsluhuje třetí až šestou vrstvu ISO/OSI modelu. Je vyhrazen pro vytváření, příjem a zpracování síťových proměnných, autorizaci, diagnostiku provozu na síti na pozadí, časovače a čítače pro organizaci řazení paketů a vyhledávání duplicit a další. Pro komunikaci s CPU1 používá Síťový buffer, pro výměnu dat s CPU3 pak Aplikační buffer.

- CPU3 - Aplikační procesor. Tento procesor spouští uživatelskou aplikaci spolu s rutinami obsaženými v jednoduchém operačním systému Neuron Chipu. Programovacím jazykem uživatelské aplikace je Neuron C[18]

O programovacím jazyku Neuron C pojednává kapitola 5. Pro přiblížení jsou zde v bodech uvedeny hlavní rozšíření oproti ANSI C:

- Zabudovaný multitasking scheduler, jenž umožňuje spouštění obslužných rutin podle událostí a priorit.

[17]MAC (Media Access Control) ... Řízení přístupu k přenosovému médiu.

[18]Neuron C je klon ANSI C optimalizovaný a rozšířený pro potřeby distribuovaných aplikací v LonWorks. Základní vlastnosti jsou uvedeny v kapitole 5.

- Různé režimy portu vstupů a výstupů, včetně jeho přímého bitového adresování (více viz. kapitola 5.2.3).
- Snadná deklarace síťových proměnných, včetně jejich typu.
- Snadná deklarace sekundového a milisekundového časovače

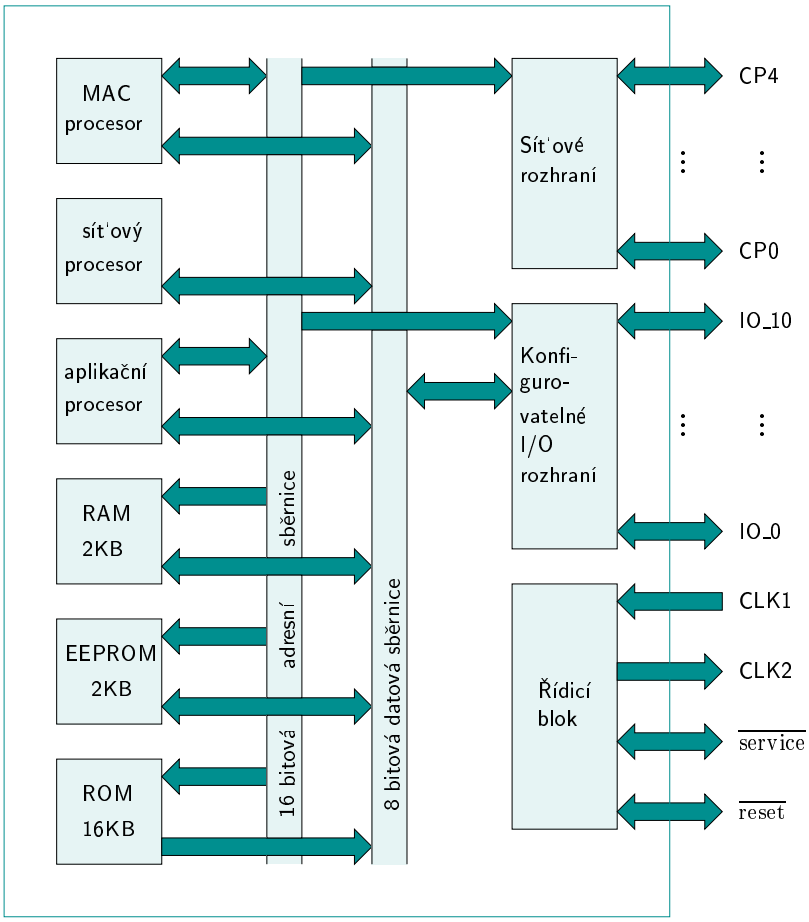
Procesory tvoří samostatné jednotky s vlastními sadami registrů, ale sdílí společnou paměť. Taktovány jsou vždy jeden po druhém, takže každý procesor běží na třetině hodinového cyklu systémového časovače.
Doba jednoho instrukčního cyklu:

$$IC = \frac{6}{f} \tag{4.1}$$

kde

IC doba instrukčního cyklu v μs.

f frekvence oscilátoru v MHz.



Obrázek 4.2: Vnitřní blokové schema procesoru Neuron Chip.

4.1 Paměť

Každý Neuron Chip procesor má vestavěny dva typy pamětí: RAM a EEPROM. Procesory řady 3120xx mají navíc paměť typu ROM, k procesorům řady 3150xx je možné připojit externí paměť. Významy a velikosti uvedených typů pamětí jsou shrnuty v následujícím přehledu.

RAM slouží k uložení pracovních dat uživatelské aplikace a LonTalk protokolu. Při výpadku napájecího napětí nebo při resetu je obsah této paměti ztracen. Konkrétně obsahuje následující položky:

- stack segment, systémová data a data uživatelské aplikace
- vyrovnávací paměť programu obsluhujícího LonTalk protokol

EEPROM umožňuje uchování obsahu i při výpadku napájecího napětí nebo při resetu. Neuron Chip má vestavěn obvod pro zápis a mazání tohoto typu paměti a tyto operace provádí firmware procesoru. Pro uživatele je práce s touto pamětí zcela transparetní. EEPROM smí být až 10.000 krát přepsána, aniž by hrozila porucha opotřebením. Časová stálost uložených dat je 10 let bez napájení. Navíc firmware procesoru zajišťuje, že zápis shodného obsahu nebude proveden, čímž se opět zvyšuje životnost této paměti. Nebezpečím je však zápis do paměti při špatné úrovni napájecího napětí. Procesor obsahuje napěťový watch-dog, ale výrobce doporučuje použití také externího zařízení. EEPROM paměť obsahuje:

- konfiguraci síťového rozhraní a adresní informace
- unikátní 48-mi bitové Neuron_ID
- kód uživatelské aplikace, některá uživatelská data

ROM mají jen procesory řady 3120xx. Zde je uložen firmware procesoru. Konkrétně obsahuje:

- kód pro obsluhu LonTalk protokolu
- tzv. Task Scheduler (více v kapitole 5.3)
- knihovny funkcí (závislé na typu procesoru)

Externí paměť je možné připojit jen k procesorům řady 3150xx. Jedná se paměť typu EEPROM, kde jsou uloženy:

- kód a data uživatelské aplikace, max. 43.000 bytů
- firmware procesoru Neuron Chip, max. 16.348 bytů

4.2 Vstupy a Výstupy

Vstupně–výstupní rozhraní Neuron Chip procesoru (I/O port) má široké možnosti komunikace se standardními zařízeními. Tvoří jej jedenáct pinů značených jako IO_0 až IO_10. Obsahuje již předpřipravené bloky, pomocí kterých se může připojit ke čtečce magnetických karet, sloužit jako vstup pro IR přijímač, připojit se ke sběrnici využívající I^2C protokol a další. Port může být také využit jako obecný paralelní nebo sériový port. Další, velmi podstatnou vlastností je možnost bitové adresace všech jedenácti pinů.

První dva piny IO_0 a IO_1 také umožňují generovat PWM[19] modulovaný signál. Tato vlastnost bude využita při realizaci zařízení (kapitola 7). PWM výstup má dva režimy: osmi a šestnácti bitový. Ty se liší v počtu kroků možných nastavení šířky pulsu, a také v rozsazích výstupních frekvencí.

Perioda a šířka pulsu výstupního signálu se určí podle vztahů

$$perioda[\text{ns}] = \frac{256 \cdot 2000 \cdot 2^{clk}}{vst_kmitočet[\text{MHz}]} \tag{4.2}$$

a

$$šířka_pulsu[\text{ns}] = \frac{výstup \cdot 2000 \cdot 2^{clk}}{vst_kmitočet[\text{MHz}]} \quad . \tag{4.3}$$

Obecná syntaxe deklarace I/O objektu v programu je uvedena v kapitole 5.2.3. Pro konkrétní případ PWM výstupu vypadá následovně:

```
<pin> [output] pulsewidth [short|long][invert][par] io_name[=init]
```

Významy obecných paramterů jsou uvedeny v kapitole 5.2.3, zde budou vysvětleny jen ty, které se týkají konkrétně PWM výstupu:

`short|long` – určuje osmi nebo šestnácti bitový režim

`invert` – otáčí fázi výstupního signálu o 180°.

`par` – vymezuje pracovní frekvenci PWM generátoru a tím zároveň určuje minimální možnou změnu šířky pulsu. Tabulky 4.1 a 4.2 udávají konkrétní hodnoty pro oba režimy. Hodnoty jsou vztaženy ke kmitočtu krystalu 10MHz.

[19] PWM (Pulse Width Modulation) – Pulsně šířková modulace

par	Frekvence [Hz]	Krok [ns]
0	19.530	200
1	9.770	400
2	4.880	800
3	2.440	1.600
4	1.220	3.200
5	610,3	6.400
6	305,1	12.800
7	152,6	25.600

Tabulka 4.1: 8-bitový režim, 255 kroků na každé frekvenci.

par	Frekvence [Hz]	Krok [ns]
0	76,29	200
1	38,16	400
2	19,06	800
3	9,53	1.600
4	4,77	3.200
5	2,38	6.400
6	1,19	12.800
7	0,60	25.600

Tabulka 4.2: 16-bitový režim, 65535 kroků na každé frekvenci.

4.3 Režimy komunikace

Komunikační rozhraní Neuron Chip procesoru má fyzicky pět pinů. Může pracovat ve třech režimech komunikace. Tabulka 4.3 shrnuje významy pinů CP0 až CP4 v jednotlivých režimech. Diferenční a Sigle-Eneded mód používají ke komu-

Pin	Režim komunikace		
	diferenční	Single-Ended	Special-purpose
CP0	RX_+ $(\leftarrow)$	RX $(\leftarrow)$	
CP1	RX_- $(\leftarrow)$	TX $(\rightarrow)$	
CP2	TX_+ $(\rightarrow)$	TX Enable $(\rightarrow)$	Bit CLK $(\rightarrow)$
CP3	TX_- $(\rightarrow)$	Sleep $(\rightarrow)$	Sleep $(\rightarrow)$ nebo Wake-up $(\leftarrow)$
CP4	Detekce kolizí $(\leftarrow)$		Frame CLK $(\rightarrow)$

Tabulka 4.3: Významy pinů komunikačního portu v různých režimech komunikace.

nikaci diferenční Manchester kódování, které se hojně používá při komunikaci po různých typech přenosových médií (např. kroucená dvoulinka, IR, RF...).

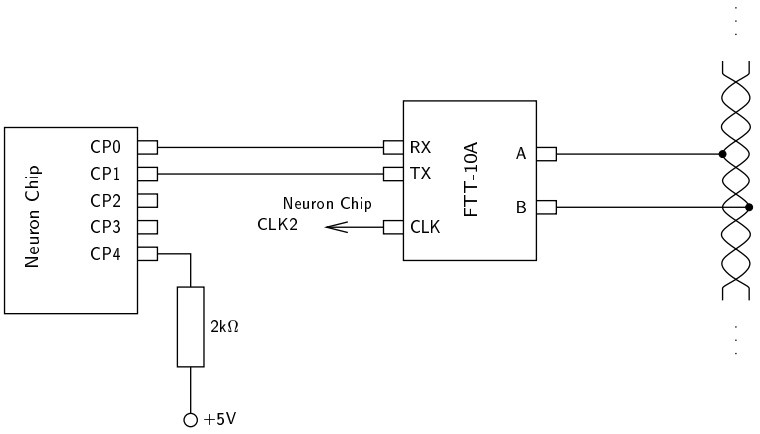
Diferenční Manchester není závislý na polaritě signálu. Princip je zřejmý z obrázku 4.4 nebo 4.6. Signál v pravidelných intervalech, které vymezují délku

jednoho bitu, mění úroveň z vysoké na nízkou a zpět. Tím se nódy zároveň synchronizují. Tyto pravidelné změny v taktu hodinového signálu vyjadřují logickou „1“. Logická nula je charakterizována ještě jednou změnou úrovně signálu uprostřed hodinového taktu.

Přepínání mezi uvedenými módy je vyřešeno softwarově – zápisem do konfiguračních registrů procesoru. K tomu je samozřejmě nutné vědět, v jakém režimu se procesor aktuálně nachází a mít příslušné rozhraní. Postup přepnutí je uveden v kapitole 7.3. Komunikační rychlost se mění v závislosti na taktovací frekvenci procesoru a použitém komunikačním médiu. Pohybuje se od 0,6kbps[20] do 2500kbps[21].

4.3.1 Single–Ended mód

Tento režim se často používá při komunikaci pomocí aktivních rozhraní, jako jsou např. FTT-10A, RS485, IR a další. Podle použitého typu se liší rychlost, topologie sítě, maximální vzdálenost a počet nódů.



Obrázek 4.3: Příklad zapojení procesoru v Single-Ended režimu, FTT-10A, kroucená dvoulinka.

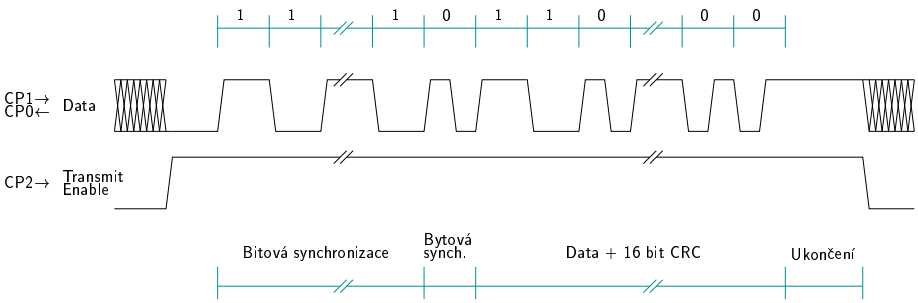
Před začátkem přenášeného paketu uvede vysílač pin CP1 (datový výstup) do nízké úrovně signálu, čímž je zaručena první změna směrem od nízké úrovně k vysoké. Zároveň je nastaven signál Transmit Enable (viz. obrázek 4.4). V intervalu jednoho bitu pak začne na pinu CP1 vysílání paketu. Samotným datům však ještě předchází několik[22] bitů ve stavu logická „1“ pro synchronizaci vysílače a přijímače. Tento blok je ukončen jedním bitem logické nula. Následuje blok dat včetně 16 bitů CRC. Paket je ukončen přidržením poslední úrovně signálu po dobu dostatečně dlouhou, aby přijímací nód zaznamenal porušení taktovací frekvence.

Pin CP3 nastavuje procesor do stavu logická nula v okamžiku, kdy přechází do stavu nízké spotřeby energie Sleep mode.

[20] kbps (**k**ilo**b**its **p**er **s**econds) – kilobity za sekundu.

[21] Tato rychlost platí jen v případě novějších procesorů s taktovací frekvencí 20MHz.

[22] Délka tohoto pole je volitelná, nejméně však čtyři bity

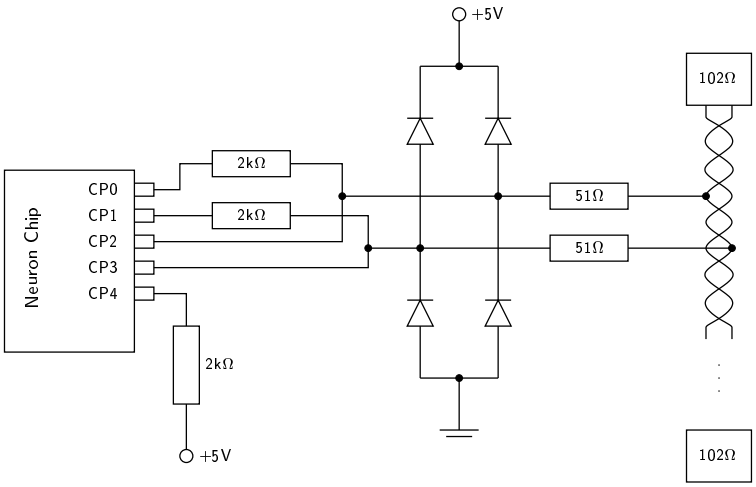


Obrázek 4.4: Časové průběhy signálu na komunikačních portech procesoru Neuron Chip v Single-Ended režimu komunikace.

Pin CP4 slouží jako vstupní, jehož pomocí může komunikační rozhraní oznámit procesoru, že během vysílání došlo ke kolizi na lince. Tuto možnost ale běžná rozhraní nevyužívají. Pak jedinou možností, jak může vysílač zjistit úspěch či neúspěch vysílání, je žádost o potvrzení doručení. Také z tohoto důvodu se nepotvrzovaná zpráva vysílá vícekrát.

4.3.2 Diferenční mód

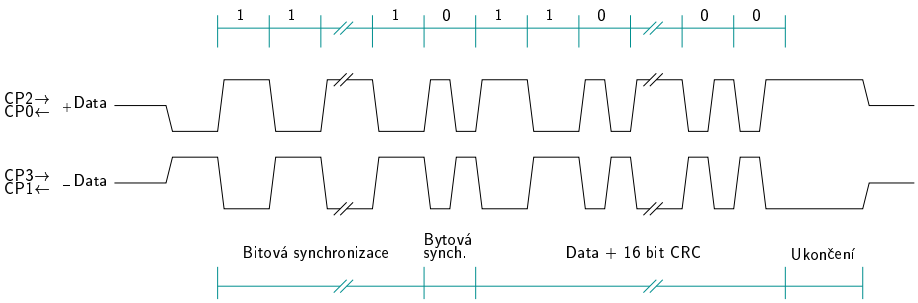
Diferenční režim komunikace je v mnoha ohledech podobný Single–Ended módu uvedenému v kapitole 4.3.1. Je ale uzpůsoben ke komunikaci pomocí rozhraní z pasivních prvků, jenž je naznačen na obrázku 4.5.



Obrázek 4.5: Příklad zapojení procesoru v diferenčním režimu, Direct Connect, kroucená dvoulinka.

Na obrázku 4.6 je zobrazen průběh vysílání jednoho paketu. Piny CP2 a CP3 zároveň vysílají shodná data, která jsou však fázově posunutá o 180°. V době, nečinnosti vysílající části jsou výstupy uvedeny do stavu vysoké impedance.

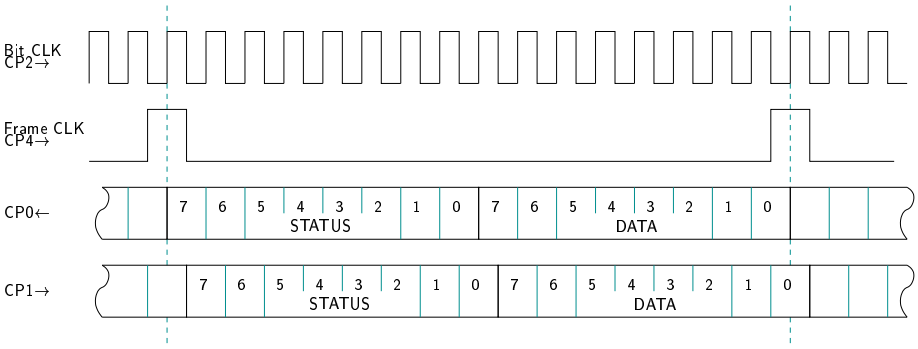
Významy jednotlivých částí časového průběhu vysílání paketu jsou shodné s vysíláním v Single–Ended režimu, jenž je uveden v předchozí kapitole.



Obrázek 4.6: Časové průběhy signálu na komunikačních portech procesoru Neuron Chip v diferenčním režimu komunikace.

4.3.3 Special–Purpose mód

V některých případech je výhodné komunikovat s vysílačem a přijímačem nekódovaně. Převod dat do potřebného formátu a přidání hlaviček, stejně jako zpětný převod přijatých dat je zcela ponechán na „inteligentním“ rozhraní. Časové průběhy signálů na jednotlivých pinech jsou zobrazeny na obrázku 4.7. CP2 a CP4 slouží k synchronizaci časovačů Neuron Chip procesoru a komunikačního rozhraní. CP0 slouží jako vysílací pin, CP1 data přijímá. Proud dat je rozdělen do 16-ti bitových bloků, kde prvních 8 bitů tvoří stavové slovo a druhých 8 bitů nese užitečná data. Významy jednotlivých bitů stavového slova jsou uvedeny v [20], kap. 9, tabulka 3.1.



Obrázek 4.7: Časové průběhy signálu na komunikačních portech procesoru Neuron Chip v Special-Purpose režimu komunikace.

5 Programovací jazyk Neuron C

Srdcem každého „inteligentního“ zařízení v síti distribuované technologie, která využívá ke komunikaci mezi jednotivými uzly protokol LonTalk, je procesor Neuron Chip. Ten ve skutečnosti ve svém pouzdře obsahuje tři procesory. Dva jsou určeny pro komunikaci v síti LON, třetí je pak vyhrazen pro uživatelskou aplikaci.

Pro vývoj aplikace, která se posléze nahraje do EEPROM paměti Neuron Chip procesoru se používá tzv. Neuron C. Je to programovací jazyk, jenž vychází z ANSI C, obsahuje však určitá rozšíření a odlišnosti. V následujících několika kapitolách jsou rozebrány ty zásadnější, kompletní popis jazyka je pak v [7]. V kapitole 5.1 jsou uvedeny některé rozdíly a rošíření od původního ANSI C, kapitola 5.2 obsahuje popis a definice nových datových typů, jako jsou SNVT proměnné, definice I/O portu a některé direktivy překladače `#pragma`. V kapitole 5.3 je nakonec popsán poněkud odlišný styl vývoje aplikací v Neuron C.

5.1 Odlišnosti od ANSI C

Programovací jazyk Neuron C se drží standardů ANSI C. Přesto však existují následující odlišnosti:

- Neuron C nepodporuje operaci s čísly v pohyblivé řádové čárce. K dispozici je ale knihovna, která tyto operace umožňuje.
- Rozdíly v bitových rozsazích typů `short int`, `int` a `long int`. Podrobnosti jsou v [6] na straně 1-12.
- Neuron C nepodporuje struktury a `union` jako parametry funkčního volání nebo návratové hodnoty.
- Neuron C nepodporuje modifikátory `volatile` a `register`.
- Nejsou podporovány ukazatele na I/O objekty, časovače ani *message tags*. Na síťové proměnné a do EEPROM paměti podporovány jsou, ale jen v omezeném rozsahu.
- Jména síťových proměnných jsou omezena na 16 znaků.
- Kód psaný v Neuron C *neobsahuje* funkci `main()`.
- Všechny reakce na podněty ze sítě nebo z I/O portu jsou vykonány ve funkcích uvozených `when()`.
- Nové třídy proměnných: `network`, `system`, `uninit`, `config` a dále `eeprom`, `far`, `offchip`, `onchip` a `ram`.

Další odlišnosti a detailnější popis výše uvedených vlastností je v [6], popř. v [7].

5.2 Nové datové typy a direktivy kompilátoru

Jak již bylo naznačeno v kapitole 5.1, Neuron C zavádí řadu nových datových typů, modifikátorů a direktiv pro překladač. O těch, které jsou nějakým způsobem zajímavé nebo byly použity pro potřeby této diplomové práce, budou pojednávat následující kapitoly.

5.2.1 Třída proměnných „network“

Pro komunikaci a vzájemné sdílení dat s ostaními nódy byla vytvořena paměťová třída `network`. Syntaxe deklarace proměnných této třídy vypadá následovně:

```
network input|output [modifikator][trida] typ [spojeni]
                                identifikator [=pocatecni_hodnota]
```

kde znak „`|`“ vyjadřuje logické „nebo“. Není tedy možné použít volbu „`input`“ a „`output`“ zároveň. Položky v hranatých závorkách jsou volitelné.
Jednotlivé části deklarace mají pak následující významy a možnosti:

input|output :

- `input` – Vyznačuje vstupní proměnnou. Zařízení bude přijímat hodnoty této proměnné a dále je zpracovávat.
- `output` – Takto deklarovaná proměnná bude výstupní. Zařízení ji bude vysílat buď automaticky nebo „na vyžádání“ (tzv. polled), podle dalších parametrů.

modifikátor :

- `sync|synchronized` – Hodnoty, kterých daná síťová proměnná postupně nabyla, musí být odeslány v nezměněném pořadí a všechny bez ohledu na vytížení procesoru. Pokud je ale hodnota změněna několikrát během kritické sekce, je odeslána jen poslední.
- `polled` – Proměnná bude odeslána pouze na žádost vzdáleného nódu.
- `sd_string` – Slouží k uložení řetězce znaků, který obsahuje slovní popis proměnné.

třída :

- `const` – Tato proměnná nemůže být změněna samotným programem. K tomu je oprávněn jen vzdálený nód či konfigurační program.
- `eeprom` – Proměnná bude uložena v paměti EEPROM. Bude tedy rezistentní vůči výpadku elektrické energie. Na druhou stranu je nutné vzít v úvahu, že počet zápisů do paměti tohoto typu je z fyzikálních důvodů omezen.

`config` – Specifikuje, že daná proměnná se bude chovat jako `const`, jež je uložena v paměti EEPROM. Používá se pro ukládání konfiguračních parametrů pomocí vzdáleného nódu nebo konfiguračního programu.

typ :
Typ proměnné může být volen ze dvou odlišných skupin. Tou první jsou standardní typy proměnných. Jsou vyjmenovány například v [6]. Druhou skupinou jsou Standard Network Variable Types, tedy SNVT proměnné. Je jim věnována samostatná kapitola 5.2.4. Největší možná velikost síťové proměnné z jedné i druhé skupiny je 31 bytů.

spojení :
V tomto poli se definují možnosti propojení síťových proměnných. Mezi možné varianty patří (ne)potvrzované, (ne)prioritní nebo (ne)autorizované spojení a další. O něco rozsáhlejší popis je uveden v [6], detailní popis všech voleb pak v [7].

identifikátor :
Jméno síťové proměnné by se mělo řídit doporučeními LonMarku. Ten definuje první tři znaky názvu podle následujícího pravidla:
`nvi`... vstupní proměnné
`nvo`... výstupní proměnné
`nci`... konfigurační parametry

počáteční_hodnota :
Hodnota, na kterou bude proměnná nastavena při své inicializaci. Podrobnější informace o výchozích hodnotách různých typů proměnných jsou uvedeny v [6].

5.2.2 Direktivy kompilátoru

Neuron C, stejně jako původní ANSI C, umožňuje vkládat do zdrojového kódu direktivy pro překladač. Ten potom příslušně upravuje styl překladu, popřípadě nastavuje speciální chování procesoru. Direktivy se uvozují klíčovým slovem `#pragma`, za nímž následuje právě jeden příkaz. Kompletní popis všech direktiv je uveden v [6]. Zde budou vypsány opět jen ty významnější.

`#pragma debug <volby>` Umožňuje rozšířené možnosti ladění programu. Verze firmware Neuron Chipu musí být 6 a vyšší. Více je uvedeno v [6].

`#pragma enable_io_pullups` Aktivuje vnitřní pull-up rezistory na pinech IO_4 až IO_7. Tím je možné eliminovat množství externích součástek. Pragma má efekt jen před inicializací I/O portu.

`#pragma enable_sd_nv_names` Způsobí, že překladač přiloží jména síťových proměnných do „Self-Documentation“ popisků. Jména proměnných se pak objeví při instalaci nódu v konfiguračním nástroji (viz. obr. 8.3). Tato pragma se smí v programu vyskytovat pouze jednou.

`#pragma num_addr_table_entries` Je jedna z více direktiv, které upravují velikost prostoru pro názvy síťových proměnných, popř. jejich počet. Tato určuje maximální velikost tabulky adres.

`#pragma read_write_protect` Způsobí uzamčení programové paměti nódu pro čtení i zápis. Jakmile bude obraz programu nahrán, nebude již možné dělat jakékoliv změny. V případě, že je třeba program změnit, je nutné nahradit procesor novým.

`#pragma set_std_prog_id` Nastavuje standardní identifikační číslo daného zařízení. To je osm bytů dlouhé a jednoznačně identifikuje typ, komunikační rozhraní, verzi software, výrobce atd. Může jej získat pouze firma, která je členem sdružení LonMark.

5.2.3 Časovače a typy I/O objektů

Neuron C obsahuje dva typy softwarových časovačů. Jedním typem je `mtimer`, který má nejmenší přírůstek 1ms. Jeho rozsah je 1 až 64000ms, tedy 64s. Druhý typ je `stimer`. Nejmenší přírůstek tohoto časovače je 1s a rozsah 1 až 65535s. V programu smí být deklarováno maximálně 15 časovačů.
Deklarace jsou následující:

```
mtimer|stimer [repeating] jmeno_casovace [=pocatecni_hodnota];
```

Významy parametrů jsou tyto:

`repeating` Pokud je tato volba uvedena, časovač se po doběhu opět sám zinicializuje. Odměřované úseky pak mají přesnou délku, i když obslužná rutina nemohla okamžitě reagovat.

`pocatecni_hodnota` Časový interval, který začne časovač ihned po inicializaci odčítat. Pokud není nic uvedeno, je nastavena nula.

Časovač se inicializuje zapisem nenulové hodnoty. Nulou se opět zastaví. Doběh se potom dá odchytit pomocí bloku `when()`. Událost, kterou je třeba odchytit, se nazývá `timer_expires(jmeno_casovace)`, jak dokládá následující příklad:

```
//inicializace casovace
mtimer repeated casovac = 100;

//obsluzna rutina
when( timer_expires( casovac ) )
{
  //telo obsluzne rutiny
}
```

Časovač uvedený v předchozím příkladu se po inicializaci sám spustí. Díky volbě `repeated` se po každém doběhu opět zinicializuje. Tělo obslužné rutiny tak bude provedeno každých 100ms.

Ke komunikaci s reálným světem procesor potřebuje vstupně–výstupní rozhraní (I/O port). Neuron Chip má k tomuto účelu na svém pouzdře vyhrazeno jedenáct pinů. Piny IO_4 až IO_7 mají softwarově konfigurovatelné pullup rezistory (viz. kapitoly 4.2 a 5.2.2). Jednotlivé části tohoto portu je možné nakonfigurovat do velkého množství režimů. Port tedy bude po správné inicializaci pracovat např. jako generátor PWM[23], I^{2}C rozhraní, čtečka magnetických karet, komunikační port pro hostitelský procesor a další. Přehled všech režimů je uveden v [6].

K práci s I/O portem slouží v Neuron C následující funkce: `io_in()`, `io_out()`, `io_select()`, `io_set_direction()`, `io_change_init()` a `io_set_clock()`. V souvislosti s I/O portem je nutné také zmínit dvě události reagující na změny vstupu. Jsou to `io_update_occurs` a `io_changes`. Přesná syntaxe a použití jsou uvedeny v [7].

Deklarace I/O objektu se provádí následovně:

```
<pin> <typ> [volby] jmeno_io_objektu;
```

`pin` specifikuje jeden z pinů I/O portu `IO_0` až `IO_10`. Jeden pin může být vstupním a zároveň výstupním objektem.

`typ` specifikuje typ I/O objektu.

`volby` je nepovinný parametr. Konkrétní možnosti jsou závislé na typu I/O.

`jmeno_io_objektu` je programátorem zvolené pojmenování I/O objektu.

Další podrobnosti jsou v [8] nebo [7].

5.2.4 SNVT proměnné

Aplikační vrstva LonTalk protokolu je založená na objektech definovaných skupinou LonMark. Tyto objekty umožňují výměnu a sdílení dat mezi zařízeními bez nutnosti znát bližší parametry konkrétních zařízení. Jedná se o Standard Network Variable Types, neboli SNVT proměnné. Je to soubor objektů, kde každý má své unikátní identifikační číslo (ID) a význam. Typem objektu je řečeno, jakou fyzikální veličinu nese, její jednotky a jaký je její přípustný rozsah. Jeho maximální velikost je 31 bytů. Může to být jednoduchá proměnná typu `int`, pole znaků nebo třeba struktura. Typ SNVT proměnné se identifikuje svým ID. Jména nejsou z kapacitních důvodů v paměti nódu uložena ani přenášena po LON.

Tak, jak se rozvíjí technologie LonWorks, se také rozrůstá soubor SNVT proměnných. Aktuální seznam je v dokumentu LonMark SNVT Master list (viz [10]), jenž je zdarma ke stažení na stránkách společnosti Echelon (www.echelon.com).

Příkladem SNVT proměnné může být `SNVT_Occupancy`. Ta má dle zmíněného dokumentu identifikační číslo 109. Má délku 1 byte a pouze dva stavy. Je bezrozměrná.

[23]Tento režim byl také využit při realizaci zařízení v této diplomové práci (viz. kap. 7.2).

Dalším typem objektů definovaných pomocí LonMark jsou Standard Configuration Properties Types, neboli SCPT. Jsou to doporučené soubory konfiguračních parametrů. Jako v případě SNVT, i pro SCPT existuje ucelený přehled pod názvem LonMark SCPT Master List (viz. [11]).

5.3 Vývoj aplikace

Společnost Echelon vyvinula pro procesor Neuron Chip programovací jazyk Neuron C. I když se jedná o jazyk odvozený z ANSI C, existují mezi oběma jazyky jisté rozdíly. Některé byly popsány v kapitole 5.1. Tato kapitola by měla ukázat styl vývoje aplikace, protože v tomto ohledu mezi oběma jazyky existují také jisté odlišnosti.

Prvním a na první pohled asi nejvýraznějším rozdílem, je absence hlavní funkce `main()`. Dalším rozdílem je nová a v Neuron C kódu velmi podstatná direktiva `when()`. S její pomocí jsou deklarovány reakce na jednotlivé události. Programátor píše pouze část kódu, která přímo souvisí s chováním vyvíjeného zařízení. Zbylá část je již hotova a je součástí vývojového nástroje NodeBuilder nebo LonBuilder. Na obrázku 5.1 je znázorněn obecný vývojový diagram aplikace pro Neuron Chip. V jeho horní části jsou znázorněny bloky inicializace a speciální blok „po resetu“, který je ve skutečnosti také uvozen direktivou `when()`, ale je vykonán pouze jednou, při startu nebo restartu aplikace. Pak následuje nekonečná smyčka, tzv. Scheduler. V ní se vybírá podle příchozích událostí (anglicky *event*) jim odpovídající reakce. Jednotlivé obsluhy událostí tvoří jakousi frontu, ve které mají některé z nich prioritní postavení. Z obrázku 5.1 je zřejmé, že to jsou dvě speciální wink[24] a offline[25] a potom ty, které jsou jako prioritní definovány uživatelem.
Zápis `when()` bloku:

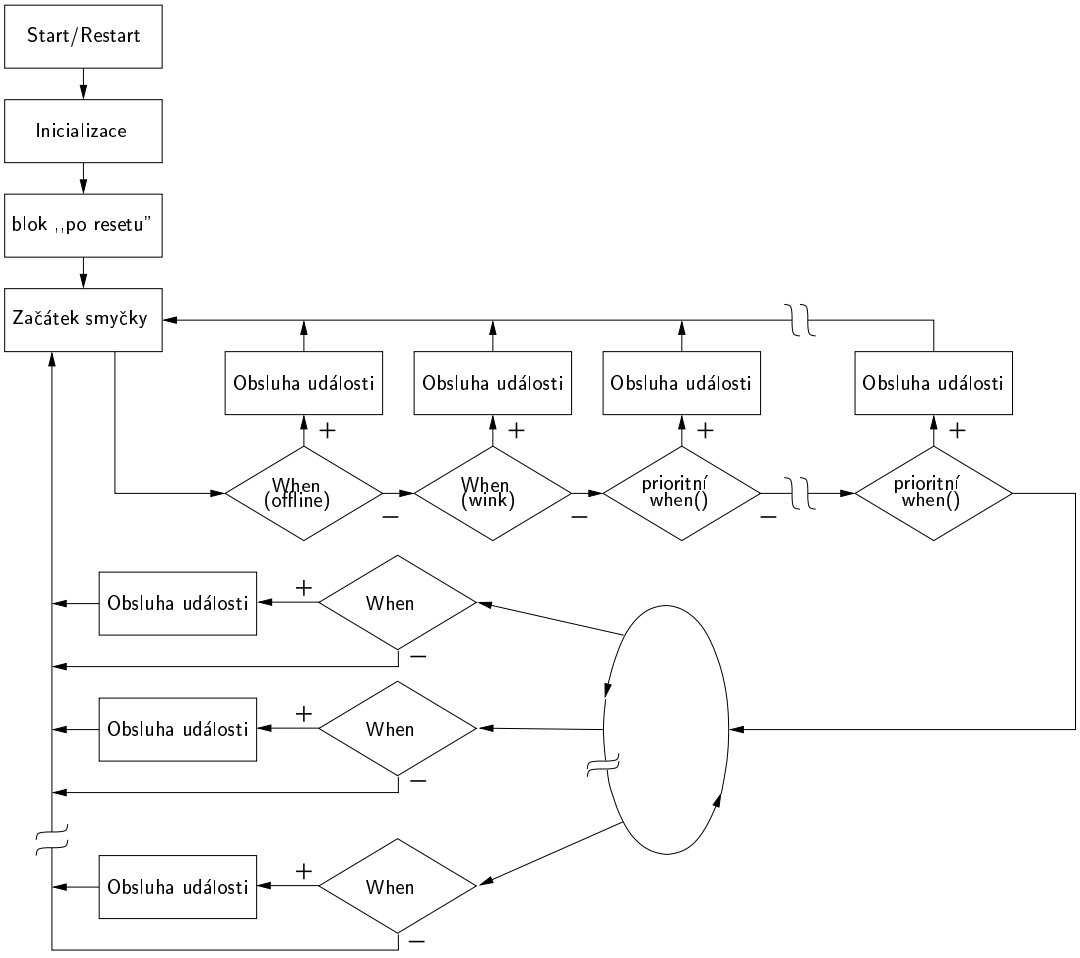
```
when ( nv_update_occurs( jmeno_sitove_promenne ) )
{
  //reakce na novou hodnotu sitove promenne
}

when ( io_changes( jmeno_io_objektu ) )
{
  //reakce na zmenu stavu io_portu
}
```

V obou uvedených příkladech je uveden konkrétní objekt (síťová proměnná, I/O objekt) u kterého je očekávána změna. Pokud bude jméno vynecháno, provede se obslužná rutina při každé změně jakékoliv síťové proměnné nebo stavu I/O objektu.

[24]Slouží pro snazší lokalizaci zařízení

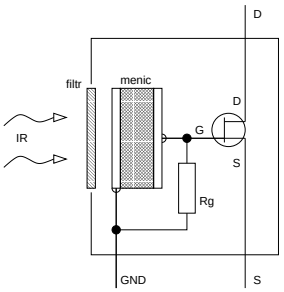
[25]Uvádí zařízení do stavu, ve kterém nepřijímá ani neysílá žádné proměnné. Zpět do normálního stavu se vrací po příkazu online.



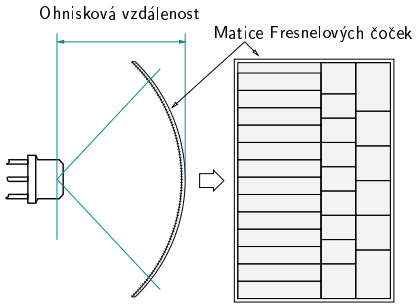
Obrázek 5.1: Obecný vývojový diagram aplikace v Neuron C.

6 Pyroelektrický senzor infračerveného záření

Každé těleso vydává určité množství elektromagnetického záření na mnoha vlnových délkách. Nás bude zajímat především infračervené záření, které je ohraničeno na jedné straně vlněním ve viditelném spektru, na druhé straně pak tepelným sáláním. Střední oblast se udává v rozmezí vlnových délek 1,5μm až 20μm. Přenos záření je také závislý na médiu. Vzduch nepřenáší všechny frekvence stejně, vznikají tak *Atmosférická okna*. Jedno takové okno je v rozmezí 6-15μm. To nás bude zajímat především.



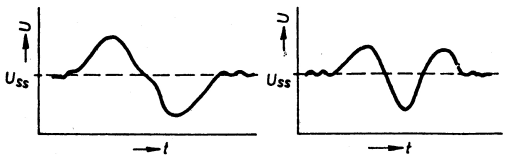
Obrázek 6.1: Základní zapojení detektoru s jedním měničem



Obrázek 6.2: Senzor je třeba umístit do ohniska pole Fresnelových čoček.

Pyroelektrický senzor infračerveného záření je pasivní detektor, jenž reaguje na *změnu* záření způsobenou např. pohybující se osobou. Základem je pyroelektrické dielektrikum, které při změně intenzity IR záření generuje napěťové pulsy. Aby se dosáhlo dostatečně krátké odezvy na změny dopadajícího záření, jsou plátky dielektrika velmi tenké, což má na druhou stranu za následek pokles přenosu asi o 6dB v oblasti kolem 3Hz.

Výstupní impedance ze samotného senzoru se pohybuje mezi 10^{12} až $10^{14}\Omega$, takže je většinou součástí detektoru vestavěný zesilovač tvořený tranzistorem FET, zapojený jako sledovač. Rezistor Rg (viz. obrázek 6) omezuje dolní mezní kmitočet, typicky 0,2Hz. Spektrální citlivost je dána materiálem filtru (viz. obr. 6). Více informací je možné získat v [2], popř. v katalogových listech firmy Nippon Ceramic Co. [23].

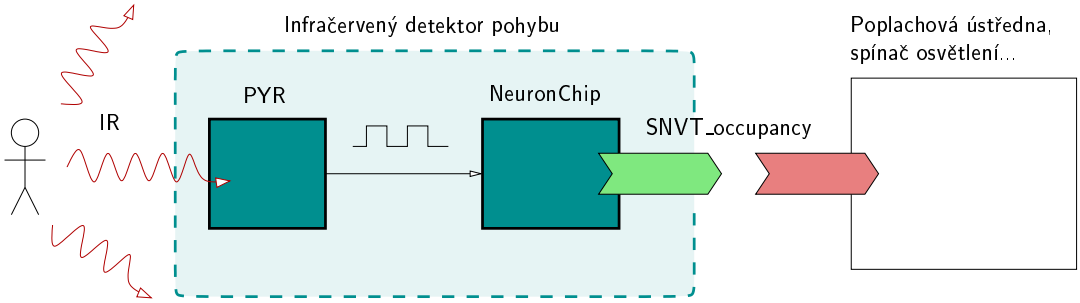


Obrázek 6.3: Základní tvar výstupního signálu detektoru s jedním senzorem(vlevo) a dvěma senzory (vpravo). Převzato z časopisu Amatérské rádio, B/5, 1992.

Aby čidlo správně fungovalo, je nutné zajistit dostatečně rychlou (nejlépe skokovou) změnu intenzity dopadajícího IR záření na citlivou část senzoru. Proto se detektor umisťuje do ohniska matice Fresnelových čoček, jak je znázorněno na obrázku 6.

7 Realizace zařízení

Součástí diplomové práce byla také realizace zařízení, komunikujícího po sběrnici LON. Bylo třeba vybrat vhodný typ a také rozhraní pro komunikaci. S ohledem na plánovaný model automatizace budovy bylo vybíráno zařízení, jež by se dalo začlenit mezi profesionální, komerčně dostupné výrobky.



Obrázek 7.1: Blokové schema funkce detektoru pohybu

Byl zvolen pasivní infračervený detektor pohybu, požadavky však byly stanoveny tak, aby se maximálním způsobem využily možnosti vzdálené konfigurace. Čidlo by tedy od okamžiku instalace v síti mělo být již minimálně lokálně udržováno, popř. nastavováno. I při rozšíření o procesor, komunikační rozhraní a další prvky, by mělo pokud možno zachovat realistické rozměry, aby bylo aplikovatelné v praxi. Dalším požadavkem byla jistá modularita tak, aby bylo

Obrázek 7.2: Fotografie realizovaného modulu IR detektoru pohybu

možné využít jednotku s procesorem a komunikací pro potřeby výuky nebo následných projektů. Funkci čidla by měla indikovat LED dioda, jako je tomu u obdobných čidel.

Na trhu existuje několik typů procesorů NeuronChip. Kriteriem výběru byla především velikost a také dosažitelnost. Proto byl vybrán typ TMPN3120FE3U [22], který má nejmenší pouzdro.

7.1 Fyzická část - Hardware

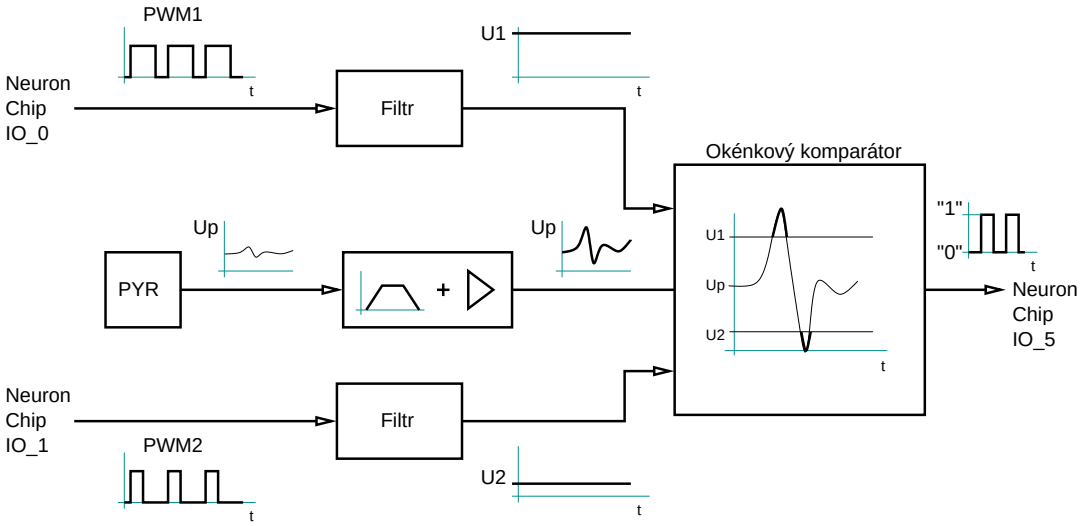
Hardware této diplomové práce sestává ze dvou samostatných částí: z analogové a procesorové. Analogová část je blíže popsána v kapitole 7.1.1. Její součástí je pyroelektrický senzor infračerveného záření a dále potřebná elektronika pro úpravu a vyhodnocení signálu z uvedeného čidla.

Procesorová část obsahuje obvod pro úpravu napětí, napěťový watch-dog, komunikační rozhraní a samozřejmě procesor NeuronChip. O této části pojednává kapitola 7.1.2.

V průběhu vývoje infračerveného čidla pohybu se ukázalo nutné vyrobit ještě přípravek pro přepnutí režimu komunikace NeuronChipu. Ten byl z výroby nastaven v tzv. „Direct Connect“ režimu, přičemž pro komunikaci přes rozhraní FTT-10A je potřeba tzv. „Single-ended“ režim. O tomto přípravku pojednává kapitola 7.1.3. Postup, jak nastavit mód komunikace je potom popsán v kapitole 7.3.

7.1.1 Analogová část - IR čidlo pohybu

Princip analogové části je schematicky naznačen na obrázku 7.3. Detailní schema zapojení je uvedeno v příloze A.1. V následujících několika odstav-



Obrázek 7.3: Blokové schema funkce analogové části.

cích budou ozřejmeny významy jednotlivých součástek na desce plošných spojů analogové části.

Protože je IR detektor IO_1 citlivý na rušení, je napájení +5V blokováno kondenzátory C_1 a C_2. Rezistor R_1 má za úkol omezit proud tímto čidlem. Na odporu R_2 se snímá výstupní signál ze senzoru. Jeho velikost je závislá na typu detektoru a pohybuje se v rozmezí 10kΩ až 47kΩ. Pro typ CSL-372 [23], použitý v této práci, udává výrobce hodnotu 47kΩ.

Signál je dále veden na vazební kondenzátor C_3, jenž stejnosměrně odděluje výstup ze senzoru a obvody pro další zpracování. Následují rezistory R_3 a R_5,

které vytváří ve společném uzlu polovinu napájecího napětí, tedy tzv. *virtuální nulu* pro zpracovávaný signál. Na tuto úroveň se musí výstupní signál z čidla posunout, aby mohl být dále střídavě zesílen nesymetricky napájeným OZ. Velikost odporu rezistorů R_3 a R_5 je třeba volit dostatečnou, aby se nezatěžoval napájecí zdroj. Na druhou stranu při příliš vysoké hodnotě těchto odporů by protékající proud byl již velmi malý a začaly by se projevovat parazitní vlastnosti součástek a rušení z okolí. Hodnota tedy byla zvolena na 100kΩ. Rezistor R_4 posouvá signál na potenciál virtuální nuly. Součástky C_3 a $R_{3,4,5}$ zároveň tvoří filtr typu horní propust. Frekvence zlomu ω_0 je při $C_3 = 10\mu F$ přibližně 1Hz.

Dále je signál přiveden na neinvertující vstup operačního zesilovače TLC272 IO_{2A} [24]. Ten signál zesiluje a zároveň tvoří pásmovou propust. Neinvertující zapojení bylo zvoleno pro svůj vysoký vstupní odpor. Kondenzátor C_4, přes který je přizeměn záporný vstup OZ, stejnosměrně odděluje zesilovač od skutečného nulového potenciálu. Díky tomu je signál z PYR čidla střídavě zesílen kolem virtuální nuly. Zároveň však ovlivňuje spolu s R_6, R_7 a C_6 frekvenční přenos celého zesilovače. Ten, jak již bylo řečeno, tvoří pásmovou propust, jejíž přenos je určen jako

$$Au = \frac{Z_1 + Z_2}{Z_1} \quad , \tag{7.1}$$

kde impedance Z_1 a Z_2 jsou

$$Z_1 = R_6 + \frac{1}{j\omega C_4} \quad , \tag{7.2}$$

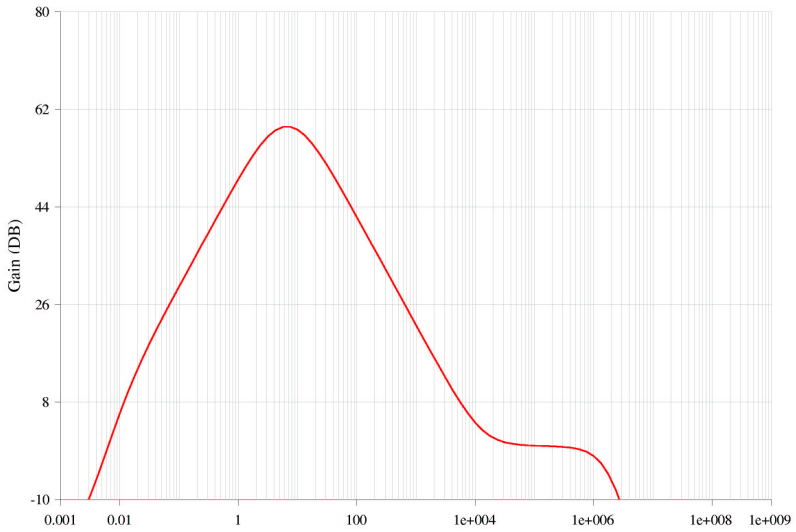
$$Z_2 = \frac{R_7 \cdot \frac{1}{j\omega C_6}}{R_7 + \frac{1}{j\omega C_6}} \quad . \tag{7.3}$$

Dosazením 7.2 a 7.3 do 7.1 získáme celkový přenos filtru. Frekvenční vlastnosti byly simulovány pomocí programu Electronics Workbench. Výsledek simulace je na obrázku 7.4.

Posledním blokem na cestě signálu od PYR čidla ke vstupu NeuronChip procesoru[26] je tzv. *okénkový komparátor*. Princip tohoto bloku je naznačen na obrázku 7.3. Jedná se o dva paralelně zapojené komparátory, z nichž každý má nastavenou jinou úroveň komparace. Výstupy jsou opět sečteny, takže pricipiálně tvoří horní a dolní napěťovou hranici, mezi nimiž se pohybuje komparovaný signál. V našem případě je jím signál z výstupu zesilovače IO_{2A}. Jakmile signál překročí jednu z hranic, překlopí se daný komparátor do druhého stavu. Ve schematu v příloze A.1 jsou to prvky označené jako IO_{3A} a IO_{3B}. Byl použit dvojitý komparátor TLC372 [25].

Pro správnou funkci celé této části je třeba generovat dvě komparační úrovně pro okénkový komparátor. To by šlo samozřejmě jednoduchým napěťovým děličem složeným ze tří rezistorů. Pro nastavení komparátoru (a tím citlivosti IR detektoru) by měl být prostřední rezistor proměnný, aby se dalo okénko měnit. To by ovšem znamenalo nutnost manuálně nastavit každé čidlo, čímž by se

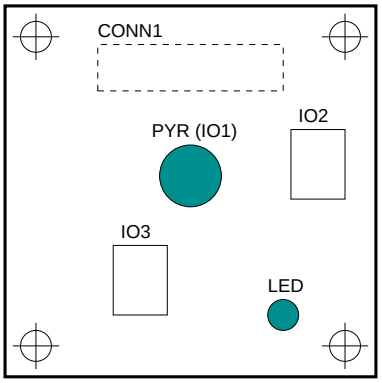
[26]NeuronChip je fyzicky umístěn na desce plošných spojů procesorové části.



Obrázek 7.4: Frekvenční charakteristika zesilovače IO_{2A} simulovaná pomocí programu Electronics Workbench.

vytrácela výhoda „inteligentního“ čidla připojeného ke komunikační sběrnici. Proto je nutné, aby zmíněné komparační úrovně generoval přímo procesor NeuronChip. Ten má, jako jeden z režimů I/O portu, schopnost generovat na pinech IO_0 a IO_1 pulsně-šířkově modulovaný signál (PWM). Oba piny tedy generují průběhy o odlišných střídách, jež jsou následně přivedeny na dva pasivní RC filtry, tvořené součástkami R_8, C_7, R_9, C_8 a R_{10}, C_9, R_{11}, C_{10}. Oba filtry jsou identické. Zlomové frekvence jsou nastaveny na 1kHz, což je desetina frekvence generovaného signálu. Díky tomu jsou na vstupech okénkového komparátoru dostatečně hladké průběhy napětí, které odpovídají středním hodnotám původního PWM signálu.

Zcela samostaně je zde také umístěna signalizační LED, která je ovládána přímo z NeuronChip procesoru.



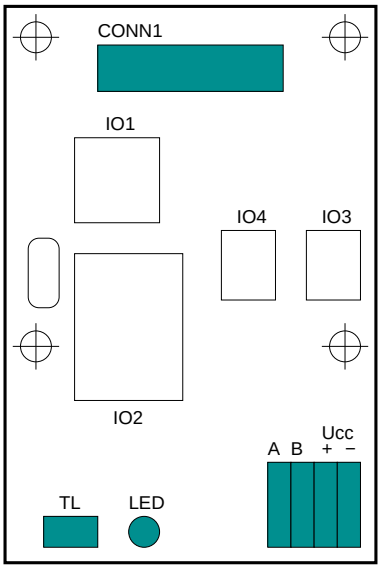
Obrázek 7.5: Blokové schema desky analogové části (Pohled ze strany součástek).

Na obrázku 7.5 je zobrazeno blokové schema fyzického rozmístění vybraných prvků. Deska plošných spojů je zde vyobrazena ze strany součástek. Pouze ko-

nektor CONN1 je umístěn z druhé strany, kvůli připojení k modulu procesorové části.

7.1.2 Procesorová část

Tato část obsahuje samotný procesor NeuronChip (IO_1) a obvody pro úpravu napájení (IO_3), napěťový watch-dog (IO_4) a komunikační rozhraní pro sběrnici



Obrázek 7.6: Blokové schema desky s procesorem NeuronChip.

LON (IO_2), jak je naznačeno v blokovém schematu na obrázku 7.6.
Pro napájení procesoru a dalších částí je třeba upravit napájecí napětí na +5V. Standardní napájecí napětí v průmyslových aplikacích je 24V. Pokud by zařízení odebíralo 50mA, bude výkonová ztráta přibližně 1W, což by způsobovalo značné zahřívání stabilizátoru. Z tohoto důvodu byl vybrán DC-DC[27] měnič MC34063 (IO_3), viz. [26]. Jedná se o výrobcem doporučené zapojení. Rezitor R_3 je snímací, poměrem rezistorů R_4 a R_5 se nastavuje velikost výstupního napětí. Cívka L_1 a kondenzátor C_9 tvoří zásobník energie pro udržování výstupní úrovně 5V. LC článek sestavený z L_2 a C_{10} tvoří výstupní filtr pro vyhlazení výstupního napětí.

Integrovaný obvod IO_4 je tzv. napěťový watch-dog TL7705A [27]. Při poklesu napájecího napětí nebo při přepětí uvede resetovací pin procesoru do aktivní úrovně[28] a procesor tak přejde do stavu *reset*. Zabrání se tím možným chybám, které by mohl generovat díky nesprávným napěťovým úrovním ve vnitřní logice.

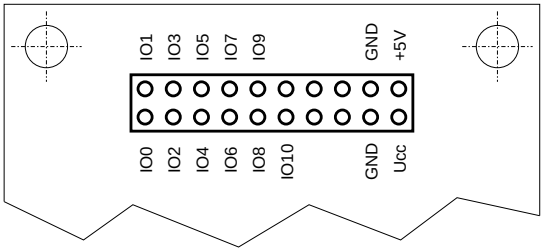
Dalším prvkem je komunikační rozhraní FTT-10A (IO_2) [17]. LonTalk protokol nemá pevně specifikovánu fyzickou vrstvu ISO/OSI modelu (viz. kapitola 3.2). V případě této práce byl zvolen právě FTT-10A, protože je tak možné toto

[27]DC-DC ... Měnič velikosti stejnosměrného napětí. Změna je podle zapojení možná oběma směry.

[28]V případě NeuronChip procesoru TMPN3120FE3U je aktivní úroveň logická 0.

čidlo připojit k již nainstalovaným komerčně vyráběným výrobkům, které jsou umístěny v laboratoři a mají shodné rozhraní. Další nespornou výhodou je možnost free-topology (viz. obr. 3.5), tedy naprosto volné vedení sítě bez nutnosti držet se příliš svazujících nároků na topologii sítě. Jedná se opět o zapojení doporučené výrobcem (viz. [17]).

Tato část byla navržena s ohledem na další využití tohoto výrobku. Na konektor CONN1 je proto vyveden kompletní I/O port procesoru, upravené napájecí napětí 5V a také neupravené napájecí napětí přivedené na vstup DC-DC měniče. Významy jednotlivých pinů jsou na obrázku 7.7.



Obrázek 7.7: Popis konektoru CONN1 pro připojení analogového modulu. Pohled ze strany součástek.

Topologie celé desky plošných spojů je znázorněna na obrázku 7.6. Tlačítko TL je tzv. *servisní tlačítko*, které např. usnadňuje instalaci zařízení v síti (viz. kapitola 4 nebo [20], [8]). Dioda LED indikuje stavy procesoru, jako jsou např. nód bez přidělené adresy domény, chyba paměti, chyba programu, stisknuté servisní tlačítko a další. Detailní informace jsou uvedeny v [15].

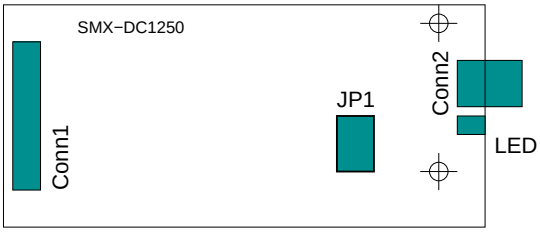
7.1.3 Přípravek SMX-DC1250

Přípravek SMX-DC1250 je modul určený pro komunikaci s procesorem Neuron Chip, který je přepnutý v diferečním režimu komunikace (viz. kapitola 4.3.2). Jedná se o jednoduché rozhraní, které umožňuje připojit na sběrnici tvořenou kroucenou dvoulinkou až 64 nodů. Podmínkou je společná zem pro všechny nody. Protože se jedná o zapojení převzaté z [20], budou zde ve stručnosti uvedeny jen základní vlastnosti a nastavení specifické pro tento konkrétní modul. Na obrázku A.3 je uvedeno schéma komunikačního rozhraní. Byly použity diody 1N4148, rezistory jsou pro maximální výkon 0,25W, třída přesnosti 5%. Na obrázku 7.8 je znázorněna samotná deska modulu.

Tato deska se pomocí konektoru Conn1 zasune do karty NSS-10, která je součástí vývojového prostředí NodeBuilder. V modulu jsou také dva montážní otvory pro uchycení. Významy prvků znázorněných v blokovém schématu karty na obr. 7.8:

Conn1 Dvouřadý, 20 pinový konektor, dutinky. Slouží k připojení modulu k desce NSS-10.

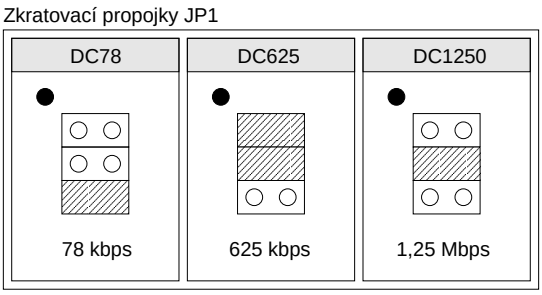
Conn2 Konektor pro připojení ke sběrnici LON.



Obrázek 7.8: Fyzické uspořádání některých prvků na desce modulu SMX-DC1250 (Pohled ze strany součástek).

LED Zelená LED dioda, indikuje vysílání na sběrnici LON.

JP1 Políčko zkratovacích propojek. Pomocí něj se dá nastavit rychlost, kterou bude karta NSS-10 komunikovat s nódem. Jednotlivé možnosti jsou vyznačeny na obrázku 7.9. Je nutné si uvědomit, že dané rychlosti se vztahují k taktovací frekvenci 10MHz.



Obrázek 7.9: Nastavení komunikační rychlosti na přípravku SMX-DC1250.

7.2 Aplikace pro NeuronChip – Firmware

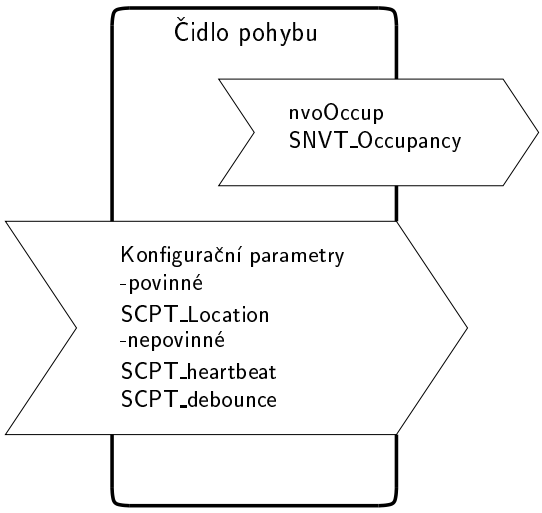
Programovacím jazykem aplikací, které pro svůj běh využívají přímo procesor Neuron Chip, je Neuron C. V případě, že by se jednalo o aplikaci, kde Neuron Chip figuruje pouze jako komunikační rozhraní (v terminologii Echelon tzv. Hosted Application), je programovací jazyk věcí vývojového prostředí procesoru, na kterém poběží daná aplikace.

V případě této práce je řídicí program dostatečně malý, aby se vešel do vnitřní EEPROM paměti Neuron Chip procesoru. Pro vývoj vlastní aplikace tedy byla zvolena první ze dvou výše uvedených možností.

7.2.1 Standardní profily

Organizace LonMark stanovuje kromě SNVT proměnných, o kterých se zmiňuje kapitola 5.2, popř. [10] nebo [6], také tzv. *funkční profily* (functional profiles). Ty stanovují standardizované chování jednotlivých typů zařízení, jako jsou například čidlo teploty, čidlo tlaku, poplachová ústředna a další. V případě detektoru

pohybu se jedná o dokument číslo 1060_10, viz. [14], který stanovuje vstupní a konfigurační SNVT proměnné podle obrázku 7.10. Významy jednotlivých ve-



Obrázek 7.10: Standardní profil čidla pohybu podle LonMark.

ličin jsou následující:

nvoOccup povinná výstupní proměnná typu `SNVT_Occupancy`. Nabývá dvou stavů, a to occupied a unoccupied. Signalizuje, zda je střežený prostor narušen. Je odesílána vždy při změně svého stavu z unoccupyed na occupyed a zpět. Doporučeno je také odesílání v časových intervalech stanovených nepovinným konfiguračním parametrem `nciHeartbeat`.

nciLocation povinný konfigurační parametr typu `SNVT_str_asc`. Jedná se o řetězec délky 0 až 31 ASCII znaků zakončený znakem `NULL`. Slouží k uložení krátké poznámky o poloze čidla.

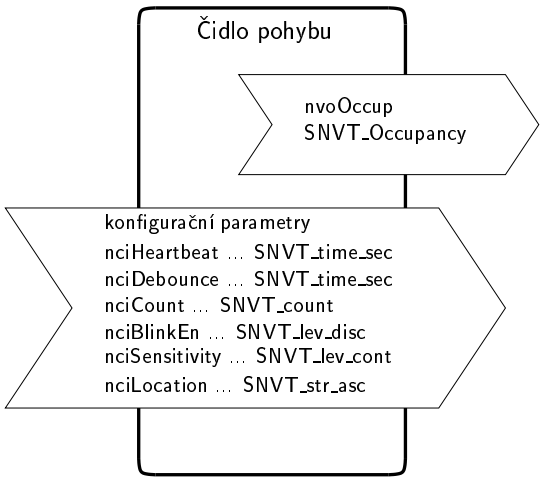
nciHeartbeat doporučený konfigurační parametr typu `SNVT_time_sec`. Udává časový interval, ve kterém je pravidelně odesílán stav `nvoOccup`. Údaj je zadáván v sekundách.

nciDebounce doporučený konfigurační parametr typu `SNVT_time_sec`. Nastavuje dobu v sekundách, za kterou je střežený prostor po narušení opět uznán jako volný.

Kromě výše uvedených povinných a volitelných parametrů jsou v této práci implementovány ještě další, rozšiřující, konfigurační proměnné:

nciSensitivity konfigurační parametr typu `SNVT_lev_cont`. Má přímou souvislost s generováním komparačních úrovní okénkového komparátoru v kapitole 7.1.1. Rozsah je 0 až 100% otevření okénka.

nciCount konfigurační parametr typu `SNVT_count`. Určuje kolik pulsů od senzoru bude přijato, než bude vyhlášen stav narušení střeženého prostoru.



Obrázek 7.11: Kompletní profil infračerveného čidla pohybu popisovaného v této diplomové práci.

Tím, že čidlo nebude reagovat na první puls přijatý od čidla, by měly být ošetřeny „falešné alarmy“, kdy je napěťová špička na výstupu pyrosenzoru vyvolána například prudkou změnou slunečního záření.

nciBlinkEn konfigurační parametr typu `SNVT_lev_disc`. V případě, že nastane porucha na vedení a dojde tím k fyzickému odpojení čidla pohybu například od poplachové ústředny nebo spínače osvětlení, nebude možné doručit síťovou proměnnou nvoOccup. Tento stav potom může signalizovat červená LED dioda zrychleným blikáním. Bude tím usnadněno a urychleno vyhledání závady a její následné odstranění. Parametrem nciBlinkEn lze výše popsanou vlastnost čidla zakázat nebo povolit.

Kompletní profil infračerveného čidla pohybu je zobrazen na obrázku 7.11. Deklarace výše uvedených proměnných vypadá takto:

```
// Vystupni promenna - udava, zda je strezeny prostor narusen
//
network output sd_string("@1|1.Occupancy state") \
                            SNVT_occupancy nvoOccup = NON_OCCUP;

// Konfiguracni promenna - Udava interval [sec], v jakem bude
//  i pri necinnosti odesilan stav cidla. Slouzi hlavne pro
//  kontrolu, zda je cidlo stale aktivni a pripojene
//
network input sd_string("@2|1.Interval for sending \
      nvoOccupancy [s]") config SNVT_time_sec nciHeartbeat = 50;

// Konfiguracni promenna - Udava casovy interval, po kterem je
//  prostor opet uznan jako volny, pokud nenastal dalsi pohyb
```

```
//
network input sd_string("@2|2.Debounce [s]") \
                          config SNVT_time_sec nciDebounce = 30;

// Konfiguracni promenna - Nastavuje se citlivost cidla.
//  Rozsah 0-100%
//
network input sd_string("@2|3.Sensitivity of the sensor [%]") \
                        config SNVT_lev_cont nciSensitivity = 0;

// Konfiguracni promenna - Udava, kolikrat musi byt zaznamenano
//  naruseni prostoru, nez je nastavena nvoOccup. Omezeni
//  falesnych alarmu.
//
network input sd_string("@2|4.Nr of pulses to \
           invoke nvoOccup [-]") config SNVT_count nciCount = 3;

// Konfiguracni promenna -Pokud je tato promenna nastavena, bude
//  cervena LED, jez signalizuje naruseni prostoru zrychlene
//  blikat, pokud se nepodari odeslat nvoOccupancy - opticka
//  signalizace preruseni spojeni
//
network input sd_string("@2|5.LED blinking if nv_update_fails \
        enabled [disc]") config SNVT_lev_disc nciBlinkEn = TRUE;

// Konfiguracni promenna -Promenna urcena k ulozeni kratkeho
//  popisu o umisteni cidla.
//
network input sd_string("@2|6.Device Location [string]") \
                                 config SNVT_str_asc nciLocation;
```

Řetězec uvnitř `sd_string()`, uvedený v deklaracích, označuje tzv. „Self-declaration string“. Ten by měl vysvětlovat význam síťové proměnné, u které je uveden. Část mezi „`@`“ a „`.`“ odpovídá pořadovému číslu objektu a proměnné podle specifikací LonMark (viz. [13]).

Významy jednotlivých modifikátorů a dalších částí deklarací jsou uvedeny v kapitole 5.2, podrobně pak v [7].

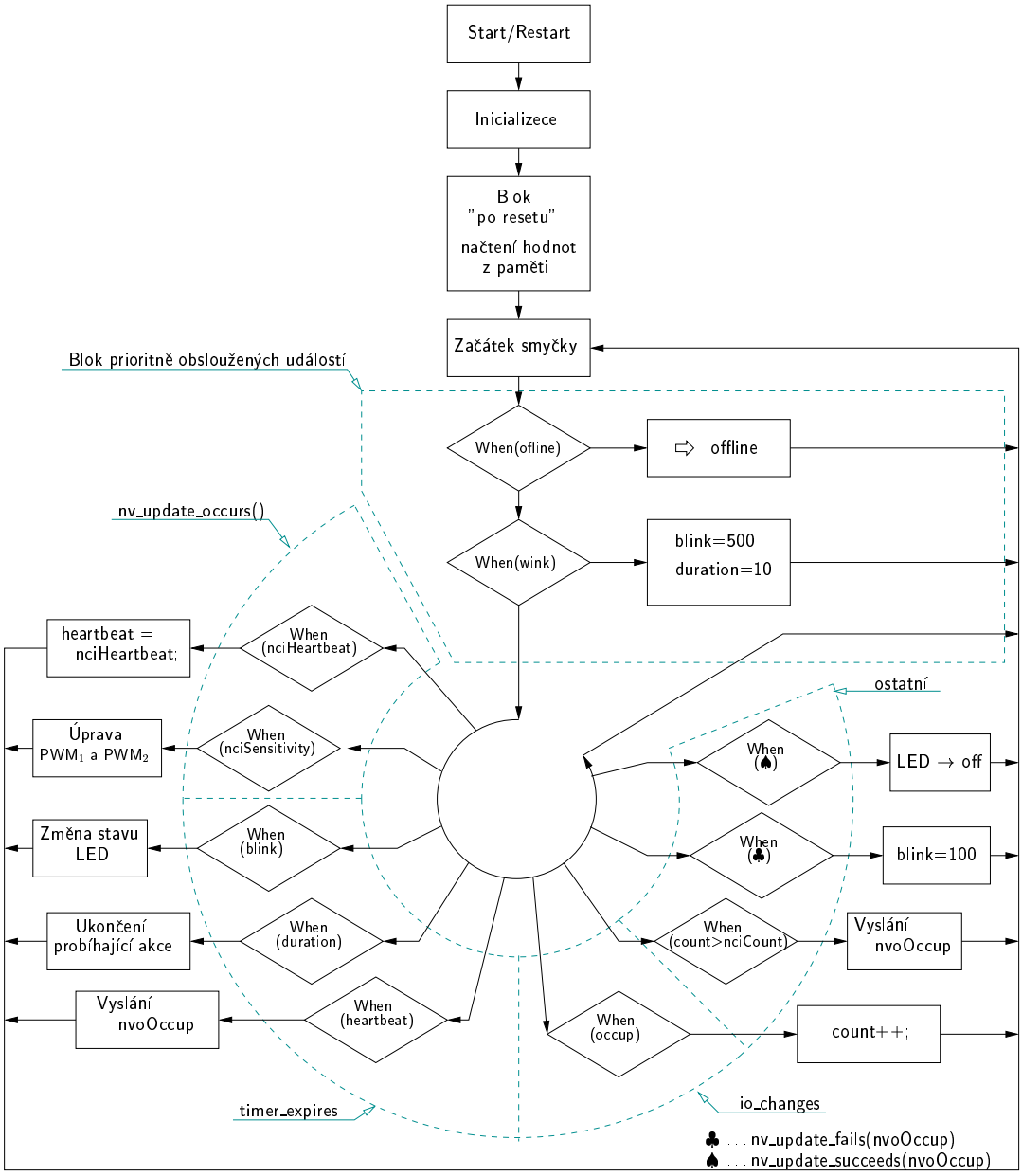
7.2.2 Obsluha událostí

Jak bylo uvedeno v kapitole 5.1, byla do jazyku Neuron C přidána nová direktiva „`when()`“. Jsou jí ošetřovány události vnitřní (např. časovače), síťové (wink, příchod síťové proměnné…) a události I/O portu (změna bitové úrovně konkrétního pinu a další). Na obrázku 7.12 je uveden zjednodušený vývojový diagram programu infračerveného čidla pohybu. V horní části tohoto schematu

je umístěn blok nazvaný „po resetu“. Jedná se také o blok uvozený direktivou `when()`, ale jeho tělo se vykoná pouze jednou, a to při startu programu. V této části jsou z EEPROM paměti načteny hodnoty proměnných a zinicializovány generátory PWM.

Následuje nekonečná smyčka, kterou pak program prochází a reaguje na prioritní a uživatelské události. Druhá jmenovaná skupina je ve schematu rozdělena do čtyř částí, podle typu události:

nv_update_occurs() signalizuje příchod nové síťové proměnné. V tomto případě jsou všechny vstupní proměnné pouze konfigurační. Většina tedy nevyžaduje další zpracování, výjimkou jsou obě uvedené `nciHeartbeat`



Obrázek 7.12: Vývojový diagram aplikace pro IR čidlo pohybu.

a `nciSensitivity`.

timer_expires() znamená, že skončil zvolený časový interval jednoho z právě aktivních časovačů. Tak, jak jsou uvedeny v diagramu, mají časovače následující významy:

blink zajišťuje (podle nastavení) různé frekvence blikání signalizační LED.

duration ošetřuje délky časově omezených akcí. Například dobu svícení signalizační LED při narušení prostoru nebo jejího blikání při wink.

heartbeat v těchto intervalech je odesílána proměnná `nvoOccup`.

io_changes() ošetřuje reakce na změny stavů vstupního I/O portu. Zde je jediná obslužná rutina pro zpracování dvoustavového výstupu z analogové části IR čidla.

ostatní je poslední část, do které jsou sloučeny zbylé obslužné rutiny. Blok v této sekci nejspodnější, zajišťuje vyslání stavu Occupied v okamžiku, kdy čítač pulsů přijatých z I/O portu přesáhne hranici danou konfigurační proměnnou `nciCount`.
Další dva bloky zajišťují blikání signalizační LED, pokud se nepodaří doručit síťovou proměnnou `nvoOccup`. A její opětovné zhasnutí, jakmile se některou z následujících zpráv opět podaří doručit.

V následujících několika odstavcích jsou uvedeny příklady konktrétních implementací jednotlivých obsluh.
Prvním příkladem je WINK, což je reakce na speciální zprávu. Tu většinou zasílá technik pomocí programu pro konfiguraci sítě při lokalizaci čidla. Zařízení na ni reaguje blikáním signalizační LED v rytmu 500ms o délce 10s.

```
// This task is called whenever the node receives a WINK message.
// It may be used by network management tools to identify nodes
// during installation.

when (wink)
{
  if( !wasOccup )
  {
    duration = 10;
    blink = 500;
  }
}
```

V první části dalšího příkladu je reakce na nedoručenou proměnnou nvoOccup. Ve druhém bloku následuje potvrzení o opětovném doručení. V první části je tedy spuštěn časovač `blink`, který generuje frekvenci blikání, v druhém bloku (pokud byla proměnná úspěšně dručena) je opět zastaven. Časovač se spouští zapsáním požadovaného času v ms.

```
//kdyz selze odeslani nvoOccup, zacne blikat signalizacni dioda
when( nv_update_fails( nvoOccup ) )
{
  if( nciBlinkEn )
  {
    wasUpdateFail = TRUE;
    blink = 100;
  }
}

//v pripade, ze se zase podari odeslat, prestane dioda blikat
when( nv_update_succeeds( nvoOccup ) )
{
  if( wasUpdateFail )
  {
    wasUpdateFail = FALSE;
    blink = 0;
    io_out( led_out, OFF );
  }
}
```

V posledním příkladu je přijata nová citlivost senzoru. Z ní jsou následně vygenerovány nové koeficienty pro generátory PWM. V poslední části jsou tyto nové koeficienty zapsány, čímž je střída PWM změněna. Citlivost se zadává v intervalu 0-100%.

```
when (nv_update_occurs(nciSensitivity))
{
  lof = (((100 - nciSensitivity/2)  * 128) / 100) + 127;
  upf = (((100 - nciSensitivity/2)  * 128) / 100);
  if( upf < 128 )
    upf = 128 - upf;
  else
    upf = 0;
  lo = (unsigned short) lof;
  up = (unsigned short) upf;
  io_out( upBound, up );
  io_out( loBound, lo );
}
```

Kompletní zdrojový kód opatřený komentáři je nahrán na přiloženém CD.

7.3 Přepnutí režimu komunikace

Některé procesory Neuron Chip jsou výrobcem předkonfigurovány v režimu „Direct-Connect“. To byl také případ procesoru použitého v této diplomové práci. V tomto režimu je procesor nastaven v diferenčním módu komunikace (viz. kapitola 4.3.2). Režimy komunikace lze nastavit softwarově, zapsáním nového nastavení do EEPROM paměti procesoru. Pro komunikaci v tomto režimu, a tedy i přepnutí do Single-Ended módu je třeba komunikační rozhraní dle obrázku 4.5 v kapitole 4.3.2. Stejným rozhraním je nutné nahradit i stávající FTM-10 Transceiver na ISA kartě NSS-10. Toto rozhraní je pod názvem SMX-DC1250 popsáno v kapitole 7.3. Stačí jej tedy mechanicky vyměnit a nastavit na něm správnou rychlost komunikace. Ta se nastaví pomocí zkratovacích propojek (viz obr. 7.9). Je třeba si uvědomit, že za základ je zde považována taktovací frekvence procesoru krystalem 10MHz a od toho si následně odvodit správnou komunikační rychlost. K samotnému překlopení (nastavení správného režimu komunikace) je zapotřebí program `nodeutil.exe` pro MS-DOS. Ten je ke stažení v sekci download na www stránkách společnosti Echelon[29]. Program je nutné spustit přímo z čistého DOSu, a pokud možno z kořenového adresáře. V jiných případech nemusí pracovat správně. Například při spuštění pod MS Windows98 nelze přijmout jakákoliv data přicházející po sběrnici LON, ale vysílal bez potíží. Samotný postup změny módu je následující:

1. Připojit NeuronChip přes rozhraní uvedené výše a připojit napájení. Je třeba mít společný potenciál země na kartě NSS-10 i procesoru.

2. Spustit program `nodeutil`

3. Přidat node do seznamu nodů v programu. Například pomocí servisního pinu (Další možnosti v [19])

4. Stisknout G (Go to node menu...) a vybrat ze seznamu nodů ten, který má být změněn

5. Stisknout T (Transceiver parameters). Nyní *opatrně*!

6. Program vypíše aktuální konfiguraci nodu (příklad je uveden v [19]) a zeptá se, zda se budou nějaké parametry měnit. Stisknout Y

7. Objeví se nabídka taktovacích frekvencí. Zvolit číslo odpovídající frekvenci krystalu na vyvíjeném zařízení.

8. Podle zadané frekvence nabídne `nodeutil` seznam přípustných režimů komunikace. Pro FTT-10A zvolit položku TP/FT-10 (v případě 5MHz taktovací frekvence to byla klávesa 2). Další typy jsou popsány v [19].

9. Následně se objeví žádost o zadání počtu opakovačů. Ponechat výchozí hodnotu.

[29]Dostupné dne 28.12.2003. Je nutná registrace, je však zdarma.

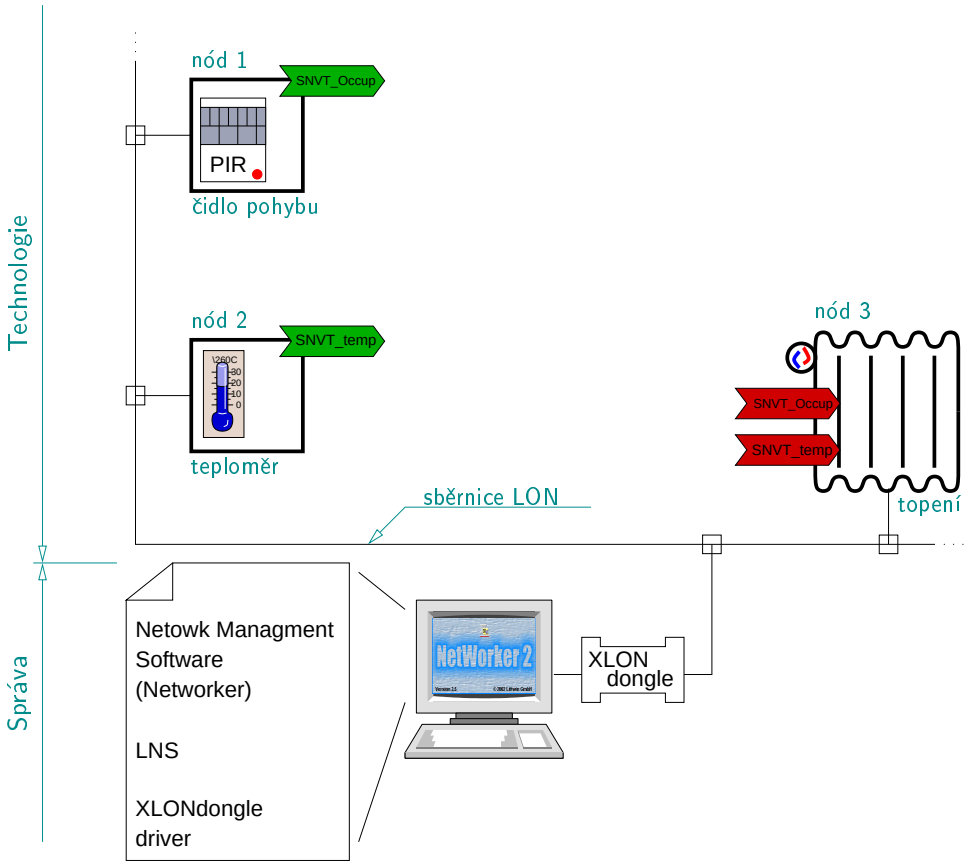
10. Poslední krok je potvrzení zapsání nových parametrů. Po zapsání nahlásí `nodeutil` chybu, protože procesor je již přepnut do jiného režimu a program s ním ztratí komunikaci.

Upozornění! Pokud je konfigurace zvolena chybně, může dojít k trvalé ztrátě komunikace. Procesory z řady 3120 jsou definitivně poškozeny, procesory 3150 je možné v některých případech zachránit vymazáním obsahu paměti.

8 Stručný přehled vývoje a integrace zařízení do sítě

Vývoj zařízení, které bude komunikovat po sběrnici LON, má dvě části (jak již bylo řečeno v kapitole 7). Tato kapitola obsahuje druhou část, tedy základní kroky pro vývoj a odladění software pro procesor Neuron Chip. V závěru je kompletní zařízení integrováno do sítě LON.

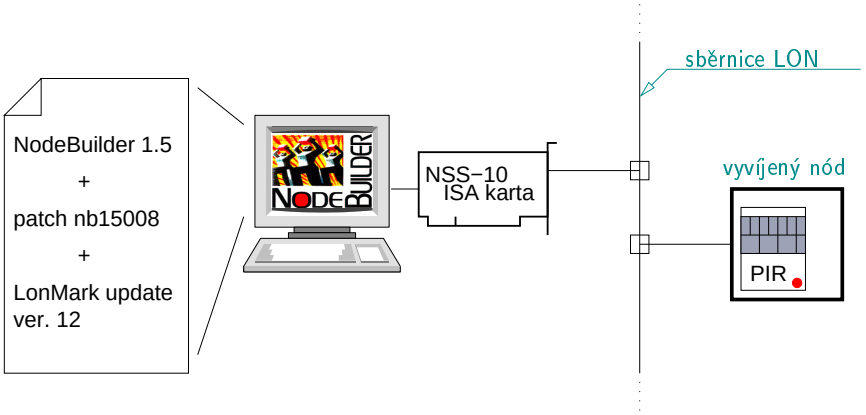
Na obrázku 8.1 je znázorněna vzorová aplikace řízení vytápění velkých výrobních hal. Jsou zde umístěny nódy s čidly teploty a pohybu. Oba nódy odesílají naměřené hodnoty jako příslušné SNVT proměnné po sběrnici LON do modulu elektrického vytápění. Ten při vyhodnocení pohybu a nízké teploty v pracovním prostoru spustí tepelný zářič.



Obrázek 8.1: Integrace čidla pohybu do sítě LON.

Pro vytvoření aplikace se používá vývojové prostředí NodeBuilder. Aktuálně (leden 2004) je na katedře řídicí techniky k dispozici verze 1.5. Ta obsahuje kartu NSS-10 (typu ISA), s jejíž pomocí lze počítač připojit ke sběrnici LON a po které se zároveň nahrává do procesoru Neuron Chip řídicí aplikace. Ovladač karty NSS-10 pracuje pouze pod Windows 3.11, Windows 95 a Windows 98. Protože verze 1.5 je poměrně zastaralá, vydala společnost Echelon několik oprav, které je nutné do nové instalace NodeBuilderu doinstalovat. Všechny opravy jsou

součástí přiloženého CD a obsahují rozšíření výběru typu procesoru a aktuální verzi databáze SNVT proměnných od asociace LonMark. Vývojové pracoviště je schematicky znázorněno na obrázku 8.2.



Obrázek 8.2: Vývoj aplikace pro čidlo pohybu pro síť LON.

Při vývoji aplikace vytvoří NodeBuilder několik typů souborů. Mezi významnější patří tyto:

`.dev` obsahuje informace o celém projektu.

`.nc` obsahuje zdrojový text uživatelské aplikace psané v Neuron C.

`.dtm` určuje typ procesoru, do kterého bude aplikace nahrána. NodeBuilder jej nevytváří, je třeba jej zvolit před nahráním aplikace do procesoru.

`.xif` obsahuje názvy a definice síťových proměnných obsažených v aplikaci. Může být později naimportován do aplikace pro konfiguraci sítě (např. Networker).

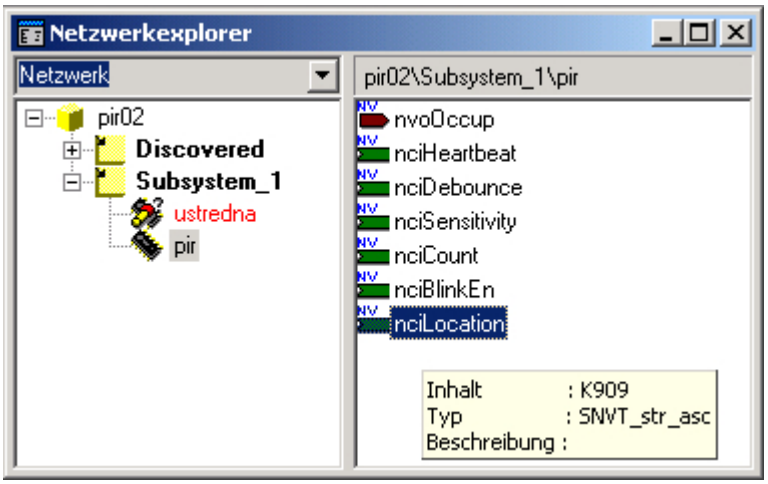
Aplikace NodeBuilder 1.5 obsahuje i jednoduchý debuger a prohlížeč síťových proměnných, což usnadňuje ladění aplikace. Další informace o balíku NodeBuilder, včetně významu ostatních souborů generovaných při překladu, jsou uvedeny v semestrální práci [5], která je ve formě naskenovaných stran umístěna na přiloženém CD.

Po odladění celého zařízení pomocí výše uvedených nástrojů pro vývoj, se zařízení integruje do sítě. Zde je možností více, jedna z nich je uvedena na obrázku 8.1. Jako rozhraní PC $\Longleftrightarrow$ LON je zde použit modul XLONdongle od společnosti DH-electronics (www.xlon.de). Ten se instaluje na paralelní port, takže je vhodný i pro přenosné počítače. Dále je nutné mít na PC nainstalovaný nástroj pro konfiguraci sítě LON. V tomto případě to byla demoverze programu Networker 2.5 (www.littwin.de), ale je možné použít také LonMaker od společnosti Echelon nebo jiný. Součástí instalace by musí být také balík LNS, který tvoří rozhraní mezi technologií LonWorks a aplikacemi na PC.

Kompletní postup integrace zařízení do sítě LON (krok za krokem) je popsán ve zprávě [4]. Ta je součástí přiloženého CD, proto zde budou popsány pouze základní kroky.

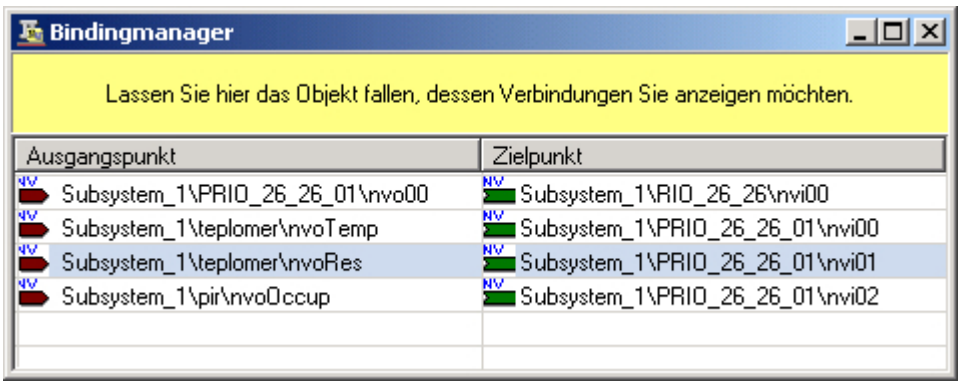
Neuron Chip má po nahrání řídicí aplikace prázdné síťové konfigurační registry. Ty jsou naplněny až v okamžiku instalace do sítě pomocí konfiguračního nástroje – Networkeru.

Po vytvoření nového projektu v Networkeru je třeba načíst jednotlivá zařízení v síti. Příklad, jak je v tomto programu zobrazeno síťové rozhraní jednoho nódu, je na obrázku 8.3. Konkrétně se jedná o infračervené čidlo pohybu vytvořené v rámci této diplomové práce.



Obrázek 8.3: Zobrazení SNVT proměnných čidla pohybu v programu Networker.

Dalším krokem je propojení příslušných síťových proměnných. To se v programu Networker provede přetažením příslušných SNVT proměnných pomocí myši do okna „Bindingmanager“. Výsledek je vidět na obrázku 8.4.



Obrázek 8.4: Propojení (bind) SNVT proměnných v programu Networker.

Tím jsou zařízení integrována v síti. Nyní je možné Networker ukončit a počítač s konfiguračním nástrojem odpojit od sítě LON.

9 Závěr

V této diplomové práci jsem se seznámil s technologií LonWorks společnosti Echelon a realizoval jsem funkční zařízení pro detekci pohybu osob v zabezpečených prostorách, které tuto technologii využívá pro komunikaci s ostatními zařízeními v síti. Připravil jsem tak půdu pro další rozvoj této technologie na katedře řídicí techniky.

Pro ověření získaných teoretických znalostí jsem vytvořil pasivní infračervené čidlo pohybu, které splňuje standard „Functional Profile: Occupancy Sensor“ asociace LonMark a zároveň jej rozšiřuje o několik konfiguračních parametrů. Čidlo je tak plně konfigurovatelné po síti. Právě tento druh zařízení jsem zvolil z důvodu možnosti zapojení do vznikající sítě komerčně vyráběných prvků automatizace budov, která by měla vzniknout v rámci následujícího projektu na katedře řídicí techniky.

Zařízení je fyzicky rozděleno na dvě části. První, analogová, obsahuje samotný pyrosenzor s potřebnou elektronikou. Druhá pak procesor a obvody pro úpravu napájecího napětí a komunikaci. Tuto koncepci jsem zvolil pro možnost snadného vývoje jiných zařízení na základě procesoru Neuron Chip. Z tohoto důvodu jsem na konektor, jenž spojuje oba moduly, vyvedl všech jedenáct pinů I/O portu a také obě napájecí napětí (napětí, které je přivedeno na vstup zařízení a také napětí upravené na hodnotu 5V). Modulární koncepce se také osvědčila při oživování a ladění výrobku.

Při vývoji jsem musel vyrobit ještě komunikační rozhraní pro diferenční režim komunikace, protože dodaný procesor měl z výroby nastaven právě tento režim. Pro komunikaci pomocí rozhraní FTT-10A je nutné, aby Neuron Chip pracoval v tzv. Single-Ended módu.

Věřím, že má práce bude přínosem v oblasti technologie LonWorks na katedře řídicí techniky a umožní další vývoj zařízení pro sběrnici LON.

Reference

[1] J. Vedral, J. Fischer:
Elektronické obvody pro měřicí techniku
Vydavatelství ČVUT, Praha, 1999

[2] *Pyroelektrické detektory infračerveného záření*
Amatérské Rádio, díl B/5, 1992, str. 163-179

[3] G. Stock, W. Meyer:
Praktische Gebäudeautomation mit LON
üthig & Pflaum Verlag GmBH & Co. Fachliteratur KG, München, Deutschland, 2003

[4] M. Linhart:
Průmyslová sběrnice LonWorks
Semestrální práce, katedra řídicí techniky, Praha, 2003, 9 str.

[5] J. Brzák:
Popis vývojového prostředí NodeBuilder Development Tool
Semestrální práce, katedra řídicí techniky, Praha, 2001, 9 str.

[6] *Neuron C Programmer's Guide*
Echelon Corp., 078-0002-02F (Rev. 6), 2002
`www.echelon.com/support/documentation/Manuals`

[7] *Neuron C Reference Guide*
Echelon Corp., 078-0140-02D (Rev. 4), 2002
`www.echelon.com/support/documentation/Manuals`

[8] *NodeBuilder User's Guide*
Echelon Corp., 078-0141-01D (Rel. 2, Rev. 6), 2002
`www.echelon.com/support/documentation/Manuals`

[9] *Introdution to the LonWorks Sustem*
Echelon Corp., 078-0183-01A (Ver. 1.0), 1999
`www.echelon.com/support/documentation/Manuals`

[10] *LonMark SNVT Master List*
Echelon Corp., Ver. 11, Rev. 2, 2002
`www.lonmark.org`

[11] *LonMark SCPT Master List*
Echelon Corp., Ver. 11, Rev. 2, 2002
`www.lonmark.org`

[12] *LonMark Layer 1-6 Interoperability Guidelines*
Echelon Corp., 078-0120-01F (Ver. 3,3), 2002
`www.lonmark.org`

[13] *LonMark Application-Layer Interoperability Guidelines*
Echelon Corp., 078-0120-01F (Ver. 3,3), 2002
`www.lonmark.org`

[14] *LonMark Functional Profile: Occupancy Sensor*
Echelon Corp., 1060_10 (Ver. 1), 1997
`www.lonmark.org/products/fprofile.htm`

[15] *Troubleshooting LonWorks Devices and Twisted Pair Networks*
LonUsers International, Nice, France, 1996
`www.echelon.com/support/documentation/docs/troubl2.pdf`

[16] *LonTalk Protocol Specification*
Echelon Corp., 078-0125-01A (Ver. 3.0), 1994

[17] *FTT-10A Free Topology Trasceiver User's Guide*
Echelon corp., 078-0156-01G (Ver. 6), 2001
`www.echelon.com/support/documentation/Manuals`

[18] *SMX Transceivers*
Echelon Corp., 003-0336-01C, 2003
`www.echelon.com`

[19] *readme z balíku nodeutil.zip*
Echelon Corp., www.echelon.com, sekce download.
Přístupné dne 29.12.2003

[20] *Toshiba Databook*
Toshiba Corp., 2001
`www.toshiba.com/taec/components/Generic/DB_neuron.shtml`

[21] *Toshiba - Product Brochure*
Toshiba Corp., Japan, 2000
`www.toshiba.com/taec/`

[22] *TMPN3120FE3M/U, katalogový list*
Toshiba America, 000707EBA2, 2000
`www.toshiba.com/taec/`

[23] *Specification of Pyroelectric Passive Infrared Sensor model CSL-372*
Nippon Ceramic Co., Ltd, drawing no 1203984, rev A, 1999

[24] *TLC272, katalogovy list*
Texas instruments Inc., Dallas, Texas, 2002
`www.ti.com`

[25] *TLC372, katalogovy list*
Texas instruments Inc., Dallas, Texas, 2002
`www.ti.com`

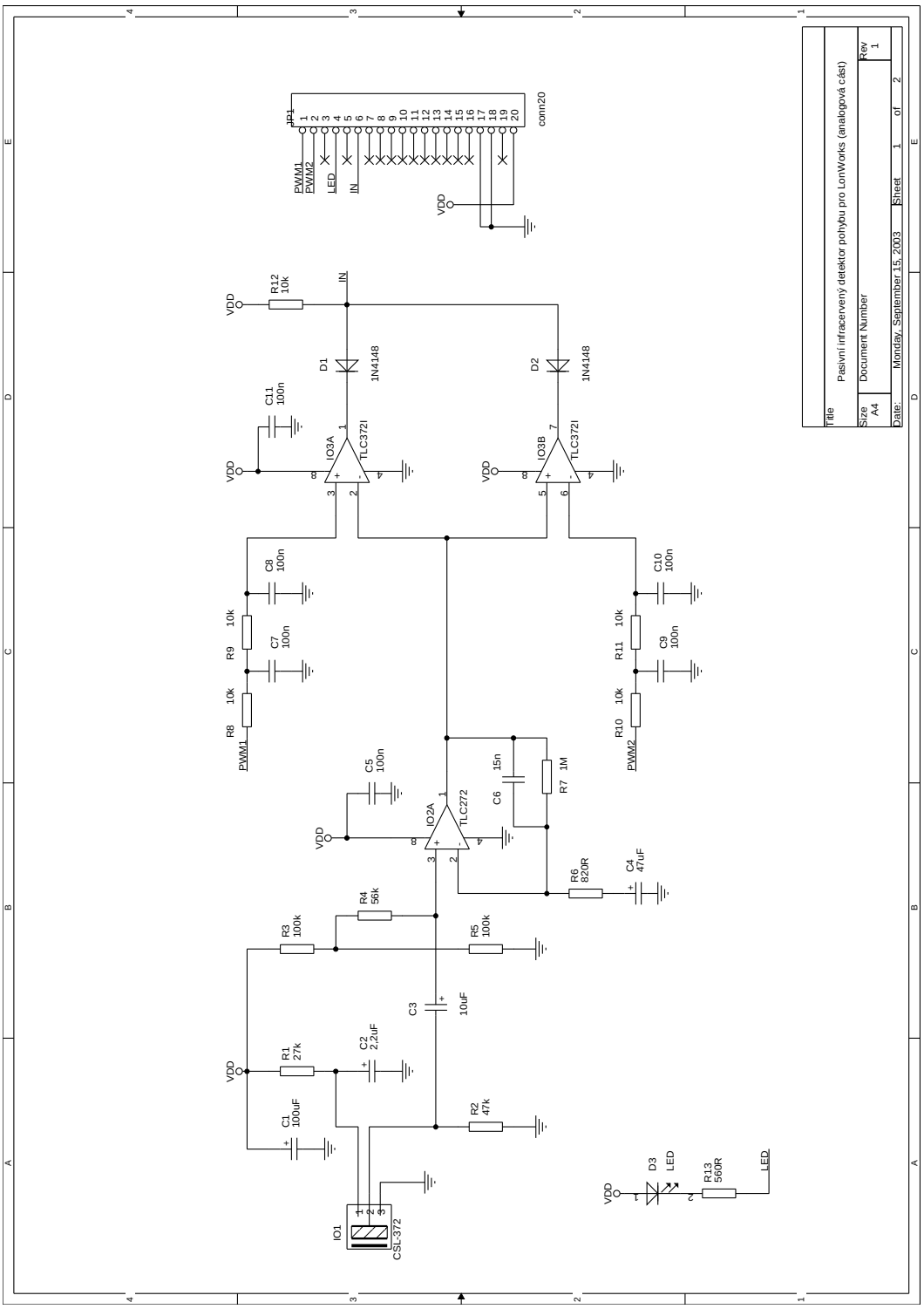
[26] *MC34063, katalogový list*
Promax-Johnton, 2002, ver. A
`www.p-johnton.com`

[27] *TL7705A, katalogový list*
Texas instruments Inc., Dallas, Texas, 1999
`www.ti.com`

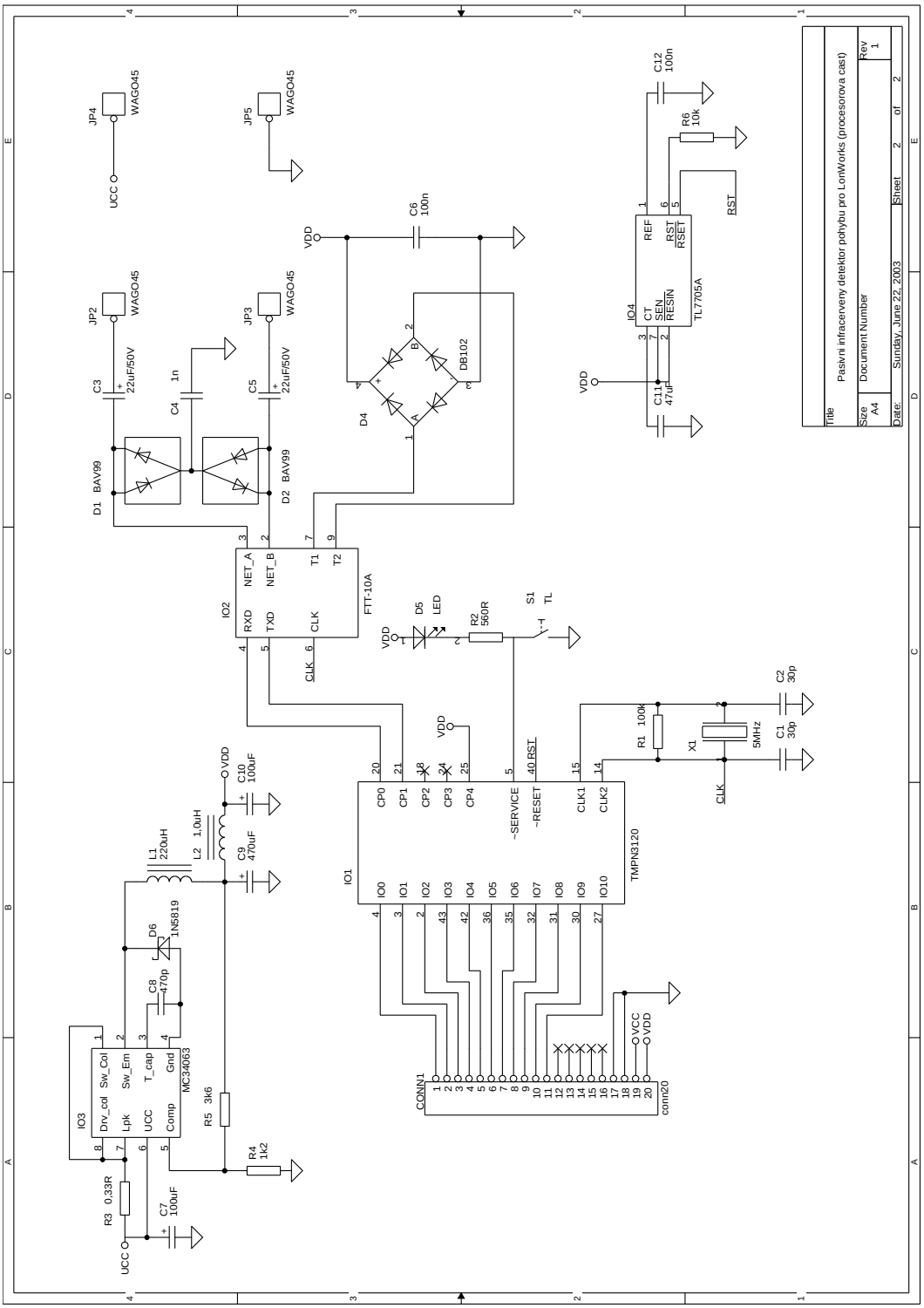
Poznámka: Publikace [6] až [22] jsou umístěny v adresáři dokumentace na přiloženém CD. Katalogové listy [23] až [27] jsou v adresáři katalogove_listy.

A Elektrická zapojení

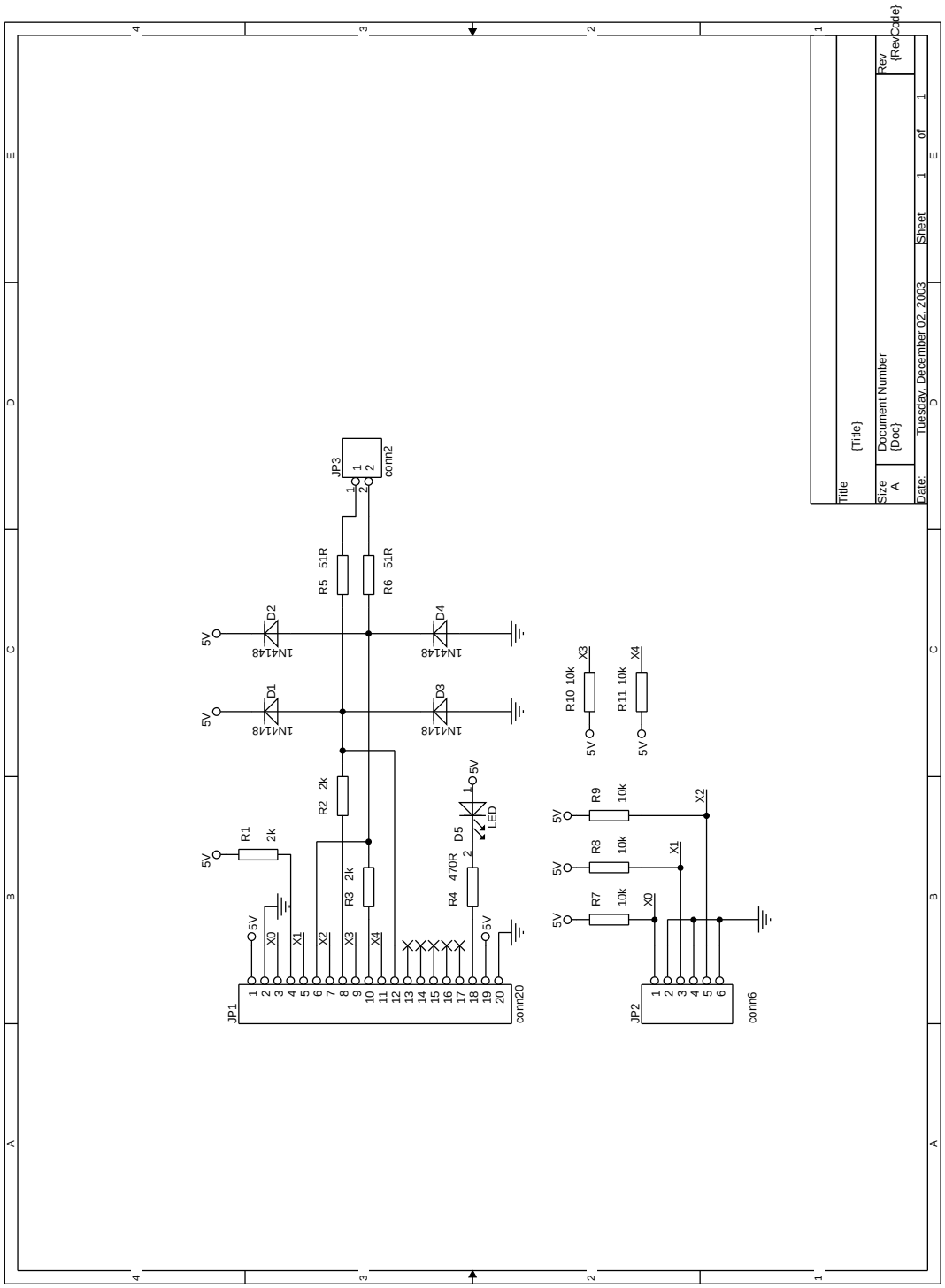
A.1 Analogová část



A.2 Procesorová část



A.3 SMX-DC1250



B Seznamy součástek

B.1 Analogová část

PIR-analogova cast

Item	Quantity	Reference	Part
1	1	C1	100uF
2	1	C2	2,2uF
3	1	C3	10uF
4	1	C4	47uF
5	6	C5,C7,C8,C9,C10,C11	100n
6	1	C6	15n
7	2	D1,D2	1N4148
8	1	D3	LED 3mm cervena
9	1	IO1	CSL-372
10	1	IO2	TLC272
11	1	IO3	TLC372I
12	1	CONN1	BL820GD
13	1	R1	27k
14	1	R2	47k
15	2	R5,R3	100k
16	1	R4	56k
17	1	R6	820R
18	1	R7	1M
19	5	R8,R9,R10,R11,R12	10k
20	1	R13	560R

B.2 Procesorová část

PIR-procesorova cast

Item	Quantity	Reference	Part
1	2	C1,C2	30p
2	2	C5,C3	22uF/50V
3	1	C4	1n
4	2	C6,C12	100n
5	2	C10,C7	100uF
6	1	C8	470p
7	1	C9	470uF
8	1	C11	47uF
9	2	D1,D2	BAV99
10	1	D4	DB102
11	1	D5	LED 3mm zluta
12	1	D6	1N5819
13	1	IO1	TMPN3120
14	1	IO2	FTT-10A
15	1	IO3	MC34063
16	1	IO4	TL7705A
17	1	CONN1	S2G20
18	4	JP2,JP3,JP4,JP5	WAGO233-508
19	1	L1	220uH
20	1	L2	1,0uH
21	1	R1	100k
22	1	R2	560R
23	1	R3	0,33R
24	1	R4	1k2
25	1	R5	3k6
26	1	R6	10k
27	1	S1	P-DT2112C (mikrotlacitko)
28	1	X1	5MHz krystal

B.3 SMX-DC1250

SMX-DC1250

Item	Quantity	Reference	Part
1	4	D1,D2,D3,D4	1N4148
2	1	D5	LED 3mm zelena
3	1	JP1	BL820GD
4	1	JP2	S2G6
5	1	L1	220uH
6	1	L2	1,0uH
7	3	R1,R2,R3	2k
8	1	R4	470R
9	2	R5,R6	51R
10	5	R7,R8,R9,R10,R11	10k

C Přehled užitých zkratek a pojmů

ANSI	American National Standards Institute Americký národní úřad pro normalizace
AuthPDU	Authentication sublayer Protocol Data Unit
CRC	Cyclic Redundancy Check Cyklická redundantní kontrola
CSMA/CD	Carrier Sense Multiple Access/Collision Detection Vícenásobný přístup s odposloucháváním nosného kmitočtu s detekcí kolizí
ISO	International Standard Organization Mezinárodní organizace pro standardizaci
OSI	Open System Interconnection Propojení definované normami ISO pro výměnu dat pomocí sedmi vrstev
LON	Local Operating Network Průmyslová komunikační sběrnice definovaná společností Echelon
LPDU	Link layer Protocol Data Unit
MAC	Media Access Control Řízení přístupu k médiu
MPDU	Media access control sublayer Protocol Data Unit
NPDU	Network layer Protocol Data Unit
PIR	Passive Infra-Red sensor Pasivní senzor infračerveného záření
PYR	Pyroelectric sensor Pyroelektrický senzor
SCPT	Standard Configuration Property Type
SNVT	Standard Network Variable Type
TPDU	Transport layer Protocol Data Unit

LonBuilder	Kompletní nástroj pro vývoj a integraci zařízení operujících na síti LON.
LonMaker	Nástroj pro integraci zařízení do sítě LON.
LonMark	Nezávislá organizace vývojářů a uživatelů LonWorks pro definování standardů pro výměnu dat mezi zařízeními.
LonTalk	Komunikační protokol pro sběrnici LON.
Neuron C	Programovací jazyk odvozený od ANSI C. Slouží pro vývoj aplikací, které budou spuštěny na procesoru Neuron Chip.
Neuron Chip	Speciální procesor používaný v technologii LonWorks.
NodeBuilder	Nástroj pro vývoj zařízení vystavěných na procesoru Neuron Chip.
Wink	Příkaz zaslaný síťovým konfiguračním nástrojem. Po jeho obdržení by mělo zařízení na sebe libovolným způsobem upozronit. Usnadňuje se tak jeho lokalizace.

D Obsah přiloženého CD

Přiložené CD obsahuje následující adresáře a soubory:

/dp_2003_martin_linhart.pdf Tento dokument ve verzi .pdf

/dokumentace adresář s dokumenty získanými z www stránek společností Echelon a Toshiba

/katalogove_listy adresář s katalogovými listy součástek, které byly použity při realizaci hardwarové části této diplomové práce

/nodebuilder adresář s aktualizacemi pro NodeBuilder 1.5

/schemata adresář schematy a návrhy plošných spojů ve formátu OrCAD 7 pro Windows.

/zdrojove_kody adresář se zdrojovými kódy aplikace pro infračervené čidlo pohybu