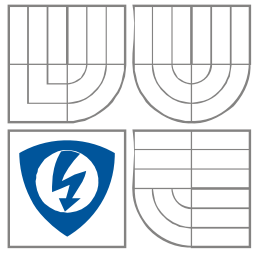
	<p>VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ BRNO UNIVERSITY OF TECHNOLOGY</p>
	<p>FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ ÚSTAV RADIOELEKTRONIKY</p> <p>FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION DEPARTMENT OF RADIO ELECTRONICS</p>

PŘÍPOSLECH K RYBÁŘSKÉMU HLÁSIČI V PÁSMU LPD
THE WIRELESS EXTENSION BOX FOR THE BITE ALARM IN LPD BAND

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Ivo Dufek

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Tomáš Frýza, Ph.D.

BRNO, 2008

Abstrakt

Cílem této práce je analýza, návrh a konstrukce funkčního prototypu tříkanálového, bezdrátového příposlechu k elektronickým signalizátorům záběru ryby v pásmu LDP 433MHz. Sada bezdrátového příposlechu se skládá z vysílače a přijímače, jejichž základem jsou hybridní moduly firmy Aurel v součinnosti s mikroprocesory AVR. Mikroprocesor ve vysílači zprostředkovává zpracování vstupních dat, jejich zabezpečení, kódování a odeslání. V přijímači se mikroprocesor stará o dekodování a vyhodnocení dat přijatých hybridním modulem přijímače a jejich následnou interpretaci, která bude optická a akustická.

Sada bezdrátového příposlechu a její konstrukce je popisována ve dvou oddělených částech. Řešení obou částí obsahuje obecný úvod, popis jednotlivých funkčních bloků a některých důležitých součástí, celkové schéma zapojení, návrh desek s plošnými spoji a popis obslužného softwaru. Nakonec následuje kapitola zabývající se vhodnými typy antén. Tato kapitola je společná pro přijímač i vysílač.

Klíčová slova

signalizátor záběru, bezdrátový příposlech, pásmo LPD, AVR, hybridní modul, přenos dat, Manchester, externí přerušení, nf zesilovač

Abstract

The main objective of this project is analysis, design and construction of three channel wireless extension box for electronics bite alarms in LPD band. Wireless extension box set consist of transmitter and receiver. Both of them are based on Aurel hybrid modules and Atmel AVR microcontrollers. Microcontroller in transmitter mediates processing input data, their coding, securing and transmitting. At the receiver side microcontroler has to interpret data from hybrid module and start correct visual and acoustic signalization.

Construction of this extension box set is described in two independent chapters. The solution contains brief introduction, description of each function block, some important components, general schematics, design of PCB and description of the firmware. The last chapter is common for transmitter and receiver and describes recommended antennas.

Keywords

electronic bite alarm, wireless extension box, LPD band, AVR, hybrid module, data transmitting, Manchester, external interrupt, AF amplifier

Bibliografická citace

DUFEK, I. Příposlech k rybářskému hlásiči v pásmu LPD. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 50 s. Vedoucí bakalářské práce Ing. Tomáš Frýza, Ph.D.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Příposlech k rybářskému hlásiči v pásmu LPD jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne 6. června 2008

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Tomášovi Frýzovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne 6. června 2008

.....
podpis autora

Obsah

1.	Úvod	7
2.	Celková koncepce	8
3.	Obvodové řešení vysílače příposlechu	10
3.1.	Blokové schéma vysílače	10
3.2.	Možnosti řešení vstupních obvodů	10
3.3.	Konkrétní řešení vstupních obvodů	11
3.4.	Použitý mikrokontrolér a jeho vlastnosti	14
3.5.	Zapojení mikroprocesoru	14
3.6.	Modul vysílače, vlastnosti a zapojení	15
3.7.	Napájení vysílače bezdrátového příposlechu	17
3.8.	Schéma zapojení vysílače.....	17
3.9.	Deska plošných spojů vysílací části	19
4.	Obslužný software mikroprocesoru vysílače	21
4.1.	Komunikační protokol	21
4.2.	Realizace software vysílače	22
4.3.	Popis jednotlivých funkcí programu vysílače	24
5.	Obvodové řešení přijímače příposlechu	26
5.1.	Blokové schéma přijímače	26
5.2.	Vstupní obvody přijímače – použitý hybridní modul	26
5.3.	Použitý mikroprocesor a jeho zapojení	28
5.4.	Nízkofrekvenční zesilovač.....	29
5.5.	Napájení přijímače bezdrátového příposlechu	30
5.6.	Schéma zapojení přijímače.....	30
5.7.	Deska s plošnými spoji přijímače	33
6.	Obslužný software přijímače	35
6.1.	Koncepce software přijímače.....	35
6.2.	Chování software přijímače v reálné situaci	36
6.3.	Popis funkce software přijímače	37
6.4.	Popis jednotlivých funkcí programu přijímače	40
6.5.	Ladění a testování software přijímače	43
7.	Vhodné typy antén	44
7.1.	Smyčková anténa.....	44
7.2.	Rezonanční anténa.....	44
7.3.	Vertikální prutová anténa umístěná na DPS	45
8.	Závěr	46
8.1.	Shrnutí dosažených výsledků.....	46
8.2.	Možnosti řešení vzniklých problémů	46
9.	Použitá literatura	48
Přílohy		49
A	Seznam součástek vysílače	49
B	Seznam součástek přijímače	50
Seznam zkratk		51

1. Úvod

Tento projekt se zabývá analýzou problematiky a návrhem tříkanálového bezdrátového příposlechu k elektronickému signalizátoru záběru pro rybáře v pásmu LPD 433 MHz.

Sportovní rybolov je v naší zemi stále velmi populární, o čemž svědčí i množství rybářů, na které můžeme denně narazit na březích našich řek, jezer a nádrží. Poslední dobou se značná část těchto lidí snaží specializovat na lov kaprů a nejlépe jen těch velkých. Taková specializace vyžaduje kromě enormního množství volného času i specifické vybavení. Loví se na zvláštní nástrahy tzv. „boilies“, používají se speciální pruty a další výbava. Výpravy za rybami bývají často velmi dlouhé (týden až 14 dnů) a bez ohledu na rozmary počasí.

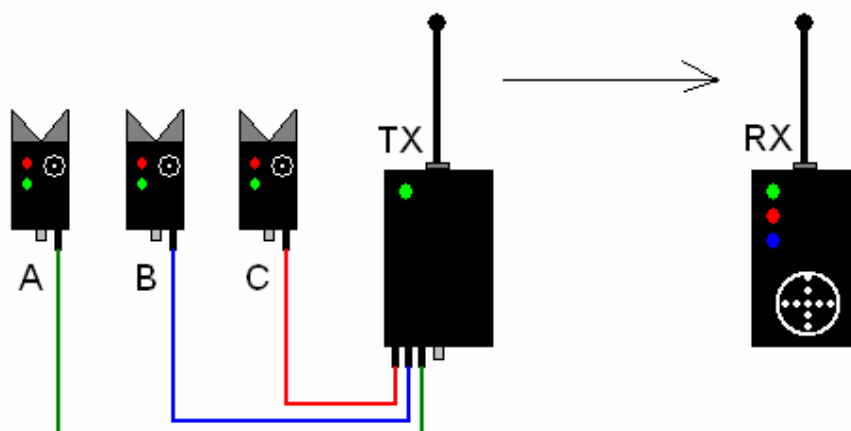
I do této oblasti si pochopitelně prorazila cestu elektronika. Skoro dvě desítky let jsou mezi rybáři rozšířeny elektronické signalizátory záběru, pracující na různých principech [9]. Toto zařízení je užitečná věc, pokud se člověk vždy pohybuje poblíž prutů, což není samozřejmě z pochopitelných důvodů vždy možné. Proto je vhodnou pomůckou právě sada bezdrátového příposlechu. Díky němu může rybář čekat na záběr v suchu svého stanu, přestože již několik hodin silně prší a pobyt venku by byl značně nepříjemný.

Sada bezdrátového příposlechu proto musí splňovat řadu požadavků. Mezi ně patří omezený dosah (několik desítek metrů), jedinečnost konkrétní sady, vysoká odolnost vůči rušení a falešným poplachům, vysoká výdrž na baterie. Sada dále musí být kompatibilní s největším možným počtem typů elektronických signalizátorů záběru a to vše za nízkou cenu.

2. Celková koncepce

V této kapitole se pokusím shrnout a rekapitulovat požadované vlastnosti a provozní parametry sady bezdrátového příposlechu, případně naznačit způsoby a možnosti jejich dosažení.

Celé zařízení bude koncipováno jako dva zcela nezávislé moduly mezi sebou propojené rádiovým signálem ve volném pásmu LPD 433 MHz.



Obr. 1: Koncepce sady tříkanálového bezdrátového příposlechu

V přijímači i vysílači budou použity hotové hybridní moduly firmy Aurel [1], používající modulaci 2 ASK (Amplitude Shift Keying) neboli OOK (On Off Keying). Tyto typy modulů byly zvoleny především kvůli jejich velmi nízké ceně. Z hlediska potřebné přenosové rychlosti moduly zcela vyhovují.

Veškerá komunikace, vyhodnocení vstupního signálu, vytvoření dat k odeslání, kódování, zabezpečení, odeslání, příjem signálu, dekódování, korekce a interpretace dat musí probíhat v reálném čase. Všechny tyto činnosti bude mít na starosti jeden mikrokontrolér osazený ve vysílači a samozřejmě druhý, který bude ovládat všechny činnosti přijímače. U obou mikroprocesorů se pokusím maximálně využít jejich provozní možnosti, jako třeba použití režimu „Power down“ během neaktivity.

K napájení prototypu vysílače poslouží čtveřice nejběžnějších a nejdostupnějších alkalických článků AAA o napětí 1,5 V. Použití tohoto druhu baterií vychází z kompromisu mezi jejich výdrží, rozměry a cenou. Výhodnější by pochopitelně bylo např. použití akumulátoru Li-ion z mobilního telefonu, který je pro provozní podmínky, kterým bude vysílač bezdrátového příposlechu vystaven, přímo konstruován. Nevýhodou by mohla být nutnost použití speciálního nabíječe v závislosti na typu akumulátoru, resp. mobilního telefonu, pro který je akumulátor určen.

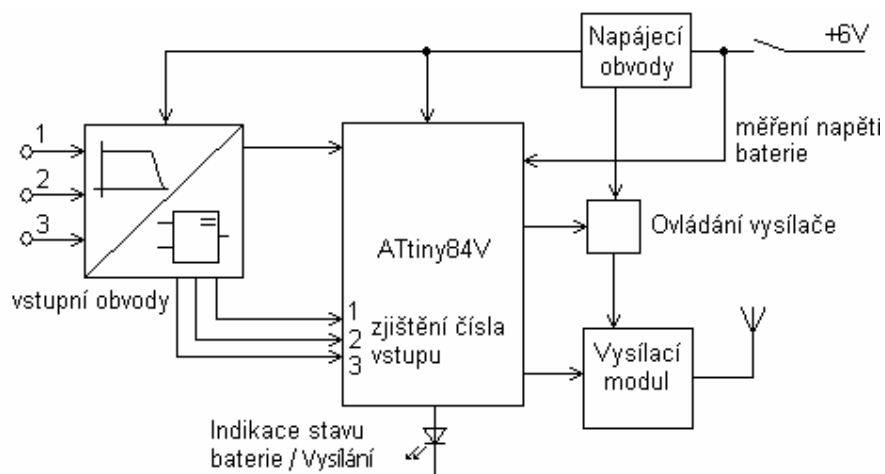
Podobně bude koncipováno i napájení přijímače. Rozdíl zde je pouze v tom, že nebude možné v režimu *stand-by* odpojit hybridní modul přijímače, tudíž zde bude vyšší klidový odběr proudu. Rozhodl jsem se proto, použít čtyři alkalické články AA

o napětí 1,5 V. Větší rozměry baterií včetně pouzdra zde nejsou na škodu, neboť bude osazen reproduktor o příkonu 0,5 W, tudíž celý přijímač bude mít větší rozměry.

Mechanické provedení přístroje není součástí tohoto projektu, neboť se jedná o prototyp, jehož uspořádání se bude dále měnit. Uvažuji ještě o několika změnách, které se promítnou do rozměrů obou částí přístroje.

3. Obvodové řešení vysílače příposlechu

3.1. Blokové schéma vysílače

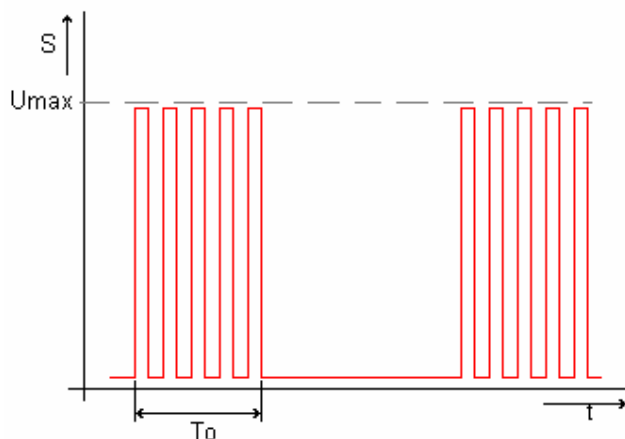


Obr. 2: Blokové schéma vysílače příposlechu

Na obr. 2 je vyobrazeno blokové schéma vysílací části sady bezdrátového příposlechu. Funkce a realizace jednotlivých bloků bude důkladně probrána v následujícím textu a to včetně obslužného software mikrokontroléru.

3.2. Možnosti řešení vstupních obvodů

Vstupní obvody vysílače sady bezdrátového příposlechu musí zvládnout správně vyhodnotit a do vhodné podoby upravit výstupní signály ze tří elektronických signalizátorů záběru. Vstupní obvody budou řešeny s ohledem na rostoucí množství typů signalizátorů, které mohou být na vstup připojeny.



Obr. 3: Typický průběh signálu na výstupu el. signalizátoru záběru [9]

Na obr. 3 je zobrazen průběh signálu, který se vyskytuje na výstupech většiny testovaných signalizátorů záběru. Úrovně signálů na výstupech různých signalizátorů se pohybovaly přibližně od 1,5V do 5V v závislosti na vnitřním zapojení. Hodnoty T_0 se pohybovaly od několika desítek ms po jejich stovky.

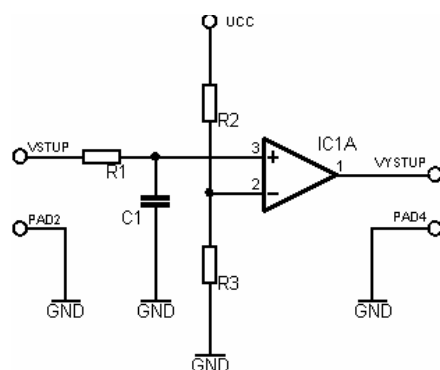
Rád bych zdůraznil, že v současné době neexistuje žádná normalizace, která by řešila problematiku elektroniky pro rybáře. Vždy se proto najdou typy signalizátorů, jejichž signál nebudou vstupní obvody navrhovaného zařízení schopny korektně zpracovat a vyhodnotit.

Úkolem vstupního obvodu je upravit signál vyobrazený na obr. 3 do podoby, kterou nejnázáve zpracuje mikroprocesor. Samozřejmě lze přivést signál na vstup mikroprocesoru v tomto tvaru, ale toto řešení by bylo náročnější na rychlost zpracování v procesoru a to není vhodné, neboť taktovací kmitočet musí být nejnížší možný z důvodu nižší spotřeby.

3.3. Konkrétní řešení vstupních obvodů

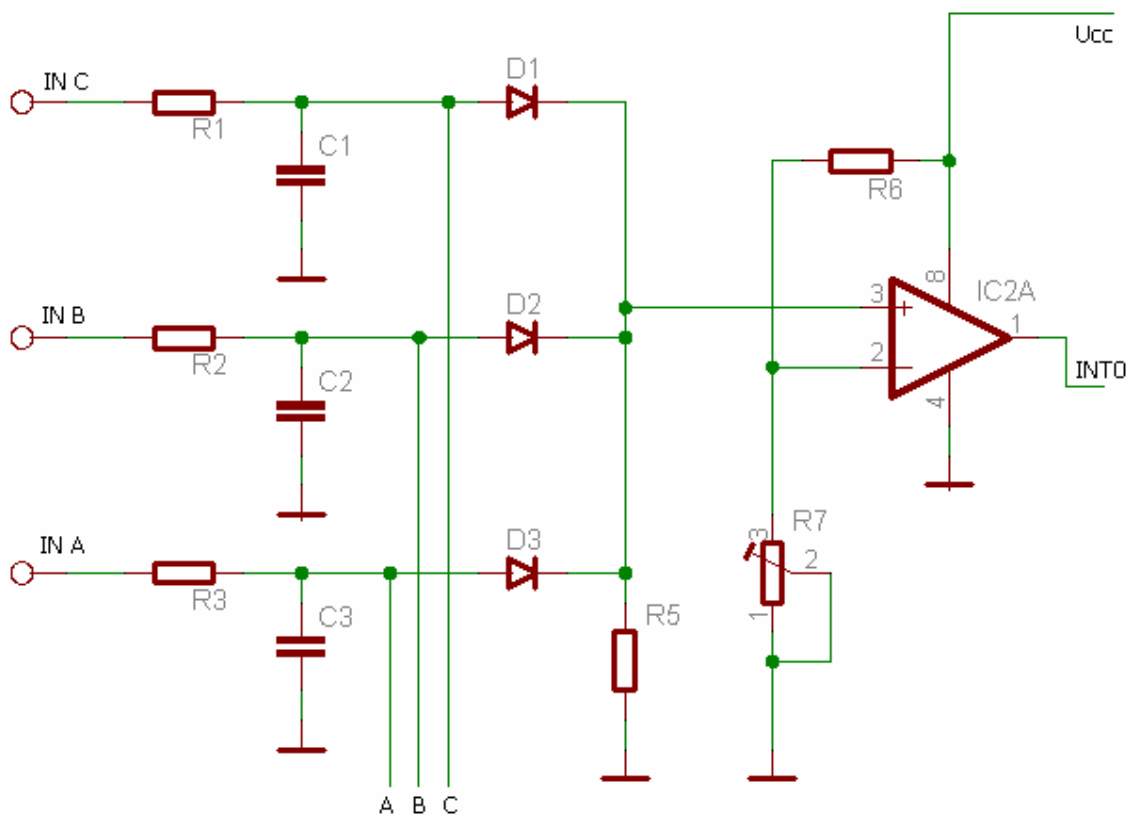
Řešení vstupních obvodů může mít několik podob. Já jsem během práce na tomto projektu otestoval následující dva způsoby řešení a pro jeden z nich se rozhodl. Vzájemně se od sebe liší množstvím signalizátorů záběru, jejichž signál dokáží zpracovat.

Signál z obr. 3 je přiveden na vstup integračního článku – dolní propusti, kde je z něj získána pouze obálka. Zároveň dojde ke zpoždění signálu a zhoršení jeho tvaru. Vhodně zvolená časová konstanta je výsledkem několika měření a následného kompromisu mezi kmitočtovým rozsahem a délkou jednotlivých úseků tohoto signálu – T_0 . Hodnota časové konstanty je 2,5 ms. Obálka signálu je následně přivedena na vstup neinvertujícího komparátoru, který zlepší tvar signálu a jeho rozkmit. V této fázi je signál připraven ke zpracování v mikrokontroléru.



Obr. 4: Schéma zapojení jednoho kanálu vstupních obvodů vysílače

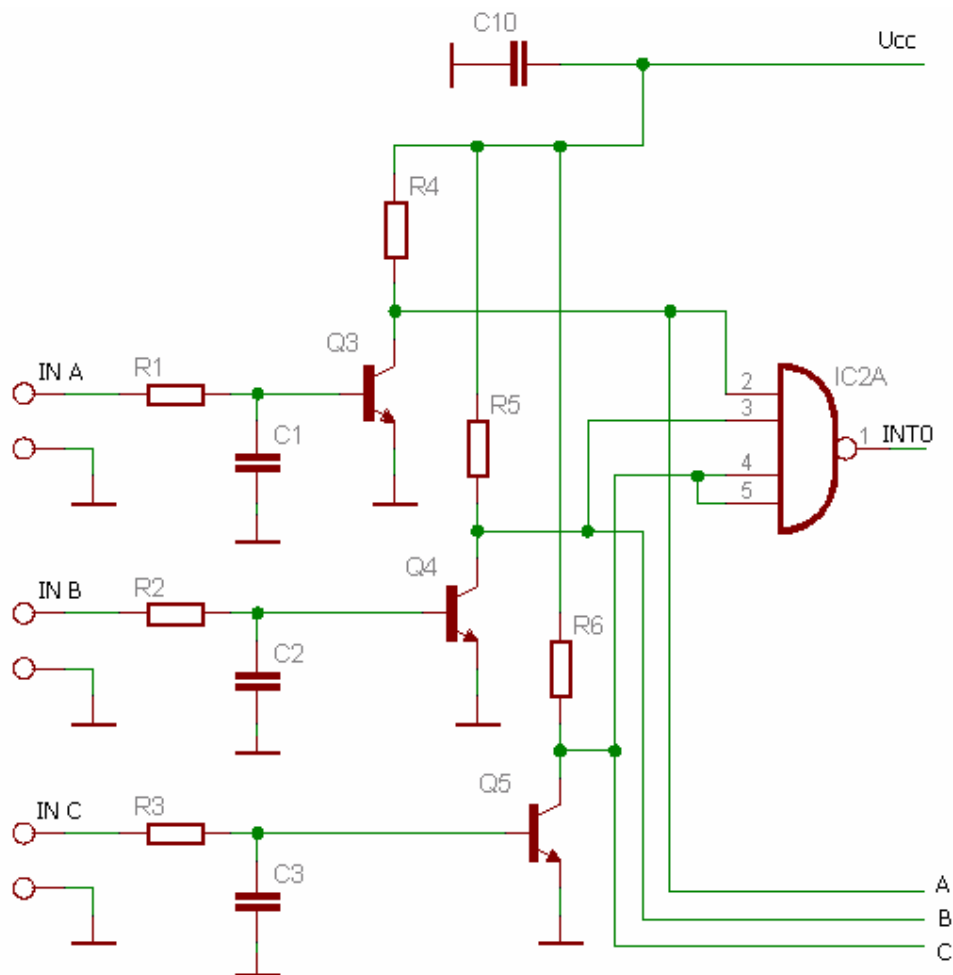
Na obr. 4 je zobrazeno schéma zapojení obvodu pro tvarování vstupního signálu, přicházejícího ze signalizátoru záběru. Obvod je tvořen integračním článkem a komparátorem v neinvertujícím zapojení.



Obr. 5: Zapojení vstupního obvodu vysílače – první řešení

Celkové zapojení vstupních obvodů vysílače příposlechu, využívající signálu od tří čidel, je na obr. 5. Zapojení je doplněno diodovým logickým obvodem OR, který má za úkol určit, právě aktivní kanály (je třeba počítat s možností záběru ryby na více udic zároveň). Tento obvod je tvořen diodami D1 – D3 a rezistorem R5. Na vstup mikroprocesoru se dostane signál vždy, ať je aktivován kterýkoli vstup a je úkolem běžícího programu, aby vyhodnotil podle signálů na vodičích A – C o které vstupy se vlastně jedná.

Tato verze vstupních obvodů bude pracovat se všemi typy signalizátorů, majících svůj výstup vyvedený přímo z mikroprocesoru nebo logického hradla, obvod s diodami, zapojený podle obr. 5 totiž potřebuje vyšší napětí, aby spolehlivě pracoval. Pokud by byl výstupní signál vyveden až ze vstupu elektroakustického měniče např. přes přepínací konektor „jack“, mohly by se vyskytnout potíže nejen s rozlišením aktivovaného vstupu, ale i s vyhlášením poplachu vůbec. Úrovně na vstupech by nemusely stačit k otevření diod a tím pádem ani k překlopení komparátoru. Je to nejzávažnější nedostatek této verze vstupních obvodů. Kompatibilitu vysílače s dalšími typy signalizátorů řeší až druhá verze vstupů.



Obr. 6: Zapojení vstupního obvodu vysílače – druhé řešení

Obrázek 6 znázorňuje schéma zapojení druhé verze vstupních obvodů vysílače. Na první pohled je patrné, že zde zůstal, jako jediná shodná součást, vstupní integrační článek RC. Zapojení dále obsahuje tři spínací stupně s tranzistory NPN, jejichž výhoda spočívá v tom, že k dostatečnému otevření není zapotřebí plného napětí log 1 TTL, ale vystačíme si s úrovní okolo 1,5 – 2V. Vzhledem k tomu, že tranzistory v zapojení se společným emitorem invertují vstupní signál, (tj. v klidu je na kolektoru log 1), je jako následující stupeň zařazeno logické hradlo NAND. V klidu je na jeho výstupu log 0 a na kontrolních výstupech A – C jsou log 1. Přejde-li signál, objeví se na výstupu hradla NAND log 1 a na některém kontrolním výstupu log 0. Takto vyřešený vstupní obvod má v klidovém stavu poměrně malý odběr (okolo 0,5 mA). Zavřenými tranzistory protéká pouze malý klidový proud I_{CE0} , tudíž hlavní část odběru má na svědomí logický obvod 4012. O další zpracování signálu se již stará program mikrokontroléru. Z hlediska celkové spotřeby elektrické energie je nejvýhodnější maximalizovat počet úkonů, řešených softwarově. Takové řešení však zvyšuje nároky kladené na mikrokontrolér, zejména si pak vynucuje užití vyššího taktovacího kmitočtu.

3.4. Použitý mikrokontrolér a jeho vlastnosti

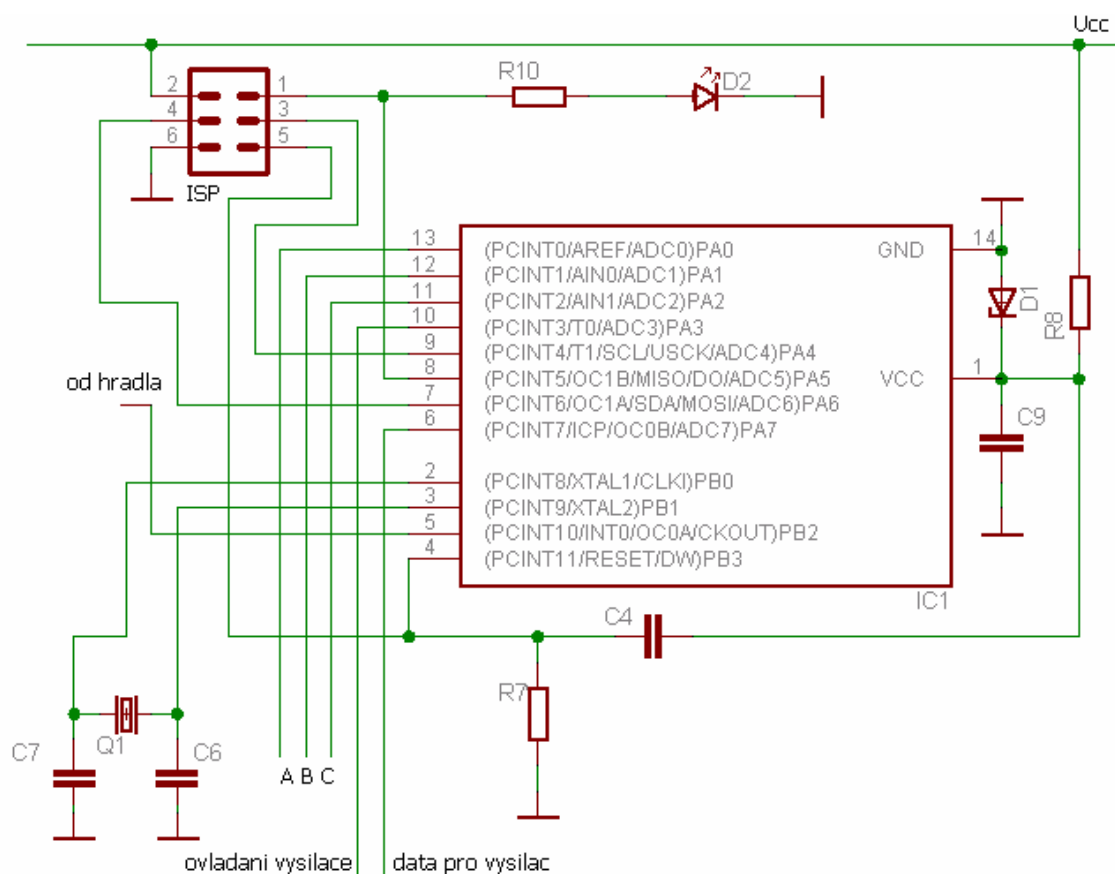
Srdcem celého vysílače je 8bitový mikrokontrolér. Běží v něm program, jenž vyhodnotí signály na vstupních pinech, zapne napájení vysílacího modulu a zabezpečí kódování a odeslání dat. Mimo to bude ještě provádět testování stavu baterií, případně nějaké další úkoly.

Po dlouhých úvahách jsem vybral jako nejvhodnější 8bitový RISC mikrokontrolér AVR firmy Atmel [2] typu ATtiny84V [3]. Jedná se o jeden z nových typů, jenž firma Atmel uvedla na trh teprve nedávno. Procesor disponuje 8 kB paměti FLASH a 512 B paměti SRAM. Mezi jeho periferiemi nalezneme také 8kanálový 10bitový ADC, integrované teplotní čidlo, vylepšený obvod RESET a další užitečné periferie. Tento obvod pracuje s napájecím napětím 1,8 – 5,5 V. Při taktovacím kmitočtu 1 MHz a napájecím napětím 1,8 V má spotřebu pouze 0,38 mA a v režimu *power down* je spotřeba dokonce jen 100 nA. Tyto procesory jsou dodávány ve 14pinových pouzdrech DIP, SOIC nebo QFN.

Po naprogramování a odladění software vysílače, jsem zjistil, že je použitý mikrokontrolér ATtiny84V zbytečně předimenzovaný a tudíž i drahý. Přeložený program zabírá něco málo přes 800 B programové paměti a 110 B paměti dat. Nabízí se zde možnost použití levnějšího obvodu z téže rodiny a to typu ATtiny24V. Tento obvod disponuje 2 kB paměti FLASH a 128 B paměti SRAM. Rozložení vývodů a obsažené periferie se plně shodují s původně použitým typem.

3.5. Zapojení mikroprocesoru

Mikroprocesor ATtiny84V je zapojen zcela standardně v souladu s katalogovým listem [3] firmy Atmel [2]. Taktovací kmitočet byl zvolen záměrně co možná nejnižší. Původním záměrem bylo taktovat mikrokontrolér kmitočtem 1 MHz. Nakonec jsem však zjistil, že bude vhodnější použít vyšší kmitočet i přesto, že stoupne proudový odběr. Resetovací obvod je tvořen rezistorem R7 a kondenzátorem C4. Minimální doba trvání impulsu na vstupu RESET je udávána 900 ns. Časovou konstantu resetovacího obvodu jsem zvolil s dostatečnou rezervou 0,1 ms. Napájecí napětí mikroprocesoru je 4,2 V a k jeho stabilizaci je použit běžný parametrický stabilizátor se Zenerovou diodou. Po dobu neaktivity vstupů resp. signálu z hradla NAND bude procesor v režimu *Power down*. Aktivace i deaktivace tohoto režimu je zajištěna v programu. Režim snížené spotřeby bude využíván co možná nejvíce. Vysílání je indikováno červenou LED diodou D2. Programování mikrokontroléru se provádí přes standardní konektor ISP. Schéma zapojení mikrokontroléru do obvodu vysílače znázorňuje obr. 7.



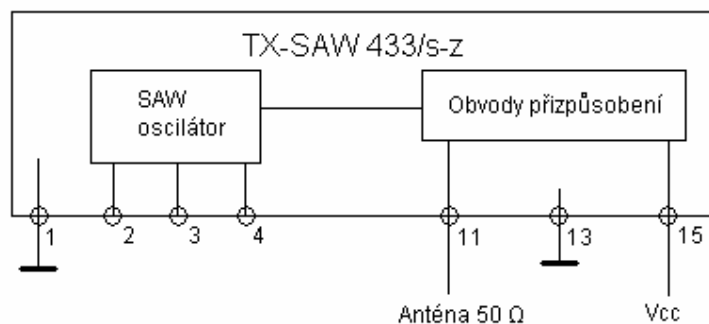
Obr. 7: Způsob zapojení řídicího mikroprocesoru

3.6. Modul vysílače, vlastnosti a zapojení

Proces samotného odvysílání dat obstarává hybridní modul TX SAW 433/s -z [4] firmy Aurel [1]. Je to nejjednodušší a nejlevnější typ, který firma nabízí. Tento modul by měl být podle údajů, uvedených v katalogovém listu plně vyhovující pro uvažovanou aplikaci.

Jedná se o homologovaný AM vysílač o rozměrech 38,1 x 13,2 x 3 mm. Zařízení používá stoprocentní modulaci tzv. amplitudové klíčování (modulace ASK neboli OOK), přičemž maximální modulační kmitočet je 4 kHz. Anténní výstup má impedanci 50 Ω . Kmitočet je stabilizován SAW rezonátorem na frekvenci 433,92 MHz. Na obr. 8 je vnitřní blokové schéma hybridního modulu vysílače.

Mimo tohoto modulu je na trhu k dostání ještě několik jiných typů od stejného nebo od jiných výrobců. Mají zpravidla lepší parametry, ale jejich cena je výrazně vyšší.



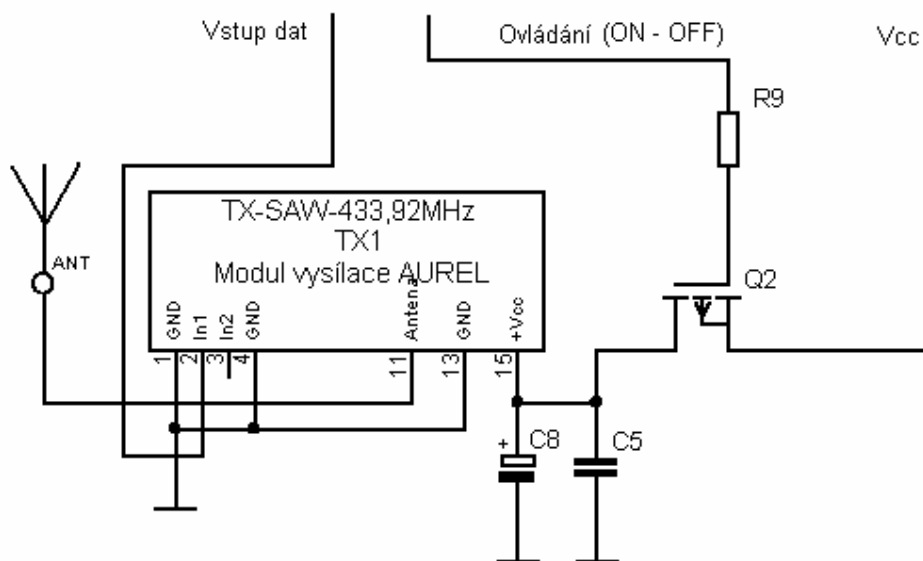
Obr. 8: Vnitřní blokové schéma modulu vysílače [7]

Maximální modulační kmitočet i výstupní výkon je závislý na napájecím napětí, jak ukazuje následující tabulka. Vývod 4 je doporučeno připojit na GND.

Tab. 1: Možnosti volby výstupního výkonu a způsoby připojení vývodů 2 a 3

Napájecí napětí Vcc [V]	Pin 2 [V]	Pin 3 [V]	Modulační kmitočet max. [kHz]	Výstupní výkon [dBm]	Spotřeba [mA]
3 – 5	0 – Vcc	Nepřipojen	3	3,5 – 8	3,5 – 7,5
5 – 8	Nepřipojen	0 – 5	4	7,5 – 10,5	3,5 – 4
8 – 12	0 – 5	Nepřipojen	4	12 – 15	7,5 – 9,5

Vzhledem k tomu, že v aplikaci bezdrátového příposlechu nepožadujeme maximální možný dosah a tím pádem ani maximální výstupní výkon, bylo zvoleno zapojení modulu v souladu s prvním řádkem Tab. 1, s datovým vstupem na pinu 2.



Obr. 9: Schéma připojení modulu TX – SAW do obvodu vysílače

Na obrázku 9 vidíme schéma zapojení modulu vysílače, odpovídající prvnímu řádku Tab. 1. Kvůli snížení spotřeby celého zařízení bude modul vysílače po celou dobu nečinnosti odpojen od napájení a připojen bude pouze v okamžiku vysílání dat, po jejichž odeslání bude opět odpojen od napájecího napětí. Tuto činnost obstarává tranzistor MOSFET s indukovaným kanálem P. Maximální přípustný proud kanálem tranzistoru je 0,25 A, při předpokládaném odběru vysílače 4 – 5 mA nehrozí v žádném případě jeho přetížení. Vysílací modul nevyžaduje stabilizaci napájecího napětí, proto je připojen přes ovládací tranzistor přímo ke svorkám napájení. V těsné blízkosti modulu jsou také umístěny blokovací kondenzátory.

3.7. Napájení vysílače bezdrátového příposlechu

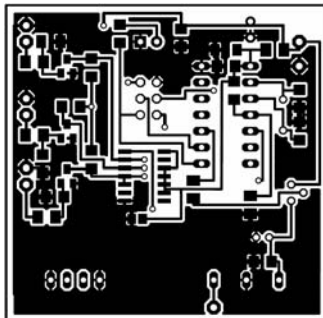
Pro napájení vysílací části sady bezdrátového příposlechu k elektronickým signalizátorům záběru byly vybrány 4 alkalické články AAA, jak bylo již výše uvedeno. Napájecí napětí je stabilizováno Zenerovou diodou na hodnotu 4,2 V. Pracovní bod Zenerovy diody byl zvolen 3 mA. Mimo Zenerovu diodu by bylo možno použít stabilizátor s obvodem LM317L nebo TL431.

3.8. Schéma zapojení vysílače

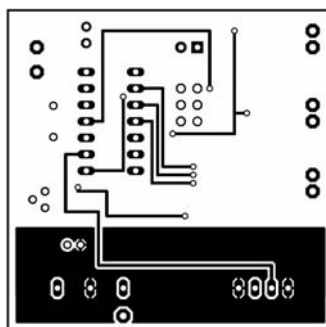
Na obrázku 10 je vyobrazeno kompletní schéma zapojení vysílací části sady bezdrátového příposlechu k elektronickým signalizátorům záběru. Schéma zapojení, stejně jako desky plošných spojů, bylo vytvořeno ve vývojovém prostředí Eagle firmy CADSoft [6]. Seznam součástek vysílače sady bezdrátového příposlechu je obsažen v příloze na konci dokumentu.

3.9. Deska plošných spojů vysílací části

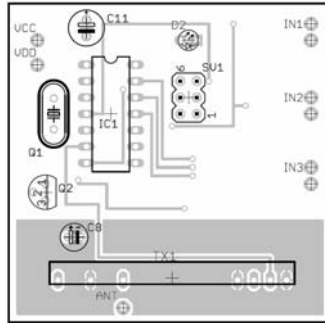
Deska s plošnými spoji vysílací části byla vytvořena ve vývojovém prostředí Eagle firmy CADSoft [6]. Je vyrobena z oboustranně plátovaného kuprexitu a má rozměry 50 x 50 mm. Velká část použitých součástek je v provedení SMD. Na desce nejsou použity žádné drátové propojky ani jakékoliv jiné prvky podobného charakteru.



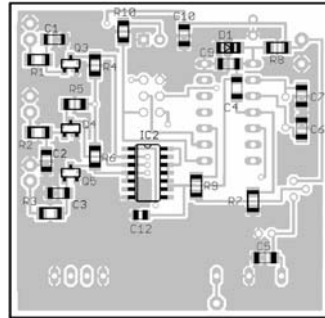
Obr. 11: Deska s plošnými spoji vysílače – strana spojů (50 x 50 mm)



Obr. 12: Deska s plošnými spoji vysílače – strana součástek



Obr. 13: Rozmístění součástek vysílače – strana součástek



Obr. 14: Rozmístění součástek vysílače – strana spojů

4. Obslužný software mikroprocesoru vysílače

Obslužný program je napsán v jazyku ANSI C, ve vývojovém prostředí AVR Studio s překladačem GCC. Než začnu popisovat provedení a funkci samotného komunikačního protokolu, je třeba vysvětlit jak bude celá komunikace probíhat, jaké budou nároky na přenosovou rychlost a datový tok.

Účelem celého zařízení je, aby doslova dokázalo přenést signalizaci záběru ryby o řádově desítky metrů dál. Tento úkol musí být proveden dostatečně rychle, hlasitě a po dobu, dost dlouhou k probuzení spícího člověka. Z výše uvedeného je jasné, že bude třeba zavést jistou minimální dobu trvání poplachu. To znamená, že i při jednotlivém impulsu ze signalizátoru záběru bude přijímač signalizovat poplach po určité době. Tuto dobu jsem stanovil na 3 sec. Pokud se nebude jednat přímo o záběr ryby, rybář dostane signál, že by bylo vhodné zkontrolovat udici.

Ze stanovené minimální doby poplachu vyplývá, možnost vysílat nižší přenosovou rychlostí při sníženém objemu dat. Jediné podstatné informace pro přijímač jsou identifikační číslo vysílače a číslo kanálu(ů), který byl aktivován. Korektně přijatá a vyhodnocená data v našem případě neznamenají nikdy nic jiného, než poplach. Jde jen o to, jak dlouho a na kterém kanálu.



Obr. 15: Formát vysílaných dat

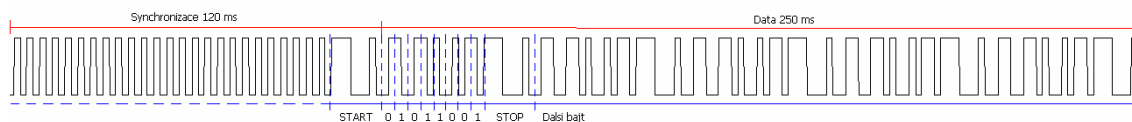
Na obrázku 15 je znázorněn formát odesílaných dat. Nejvyšší 3 bity v bajtu tvoří identifikátor aktivního kanálu. Existuje celkem 7 variant (8. varianta 000 nenastane, nebyl by vyvolán poplach – klidový stav). Dále následuje 5bitové identifikační číslo, které by mělo být jedinečné pro každou sadu. V našem případě máme k dispozici pouze 32 variant, což je velmi málo, ovšem pro účely testování je to hodnota dostatečující. Pokud by se zařízení mělo reálně vyrábět, bylo by vhodné použít aspoň 16 bitové identifikační číslo sady.

Přenosová rychlost je nastavena na hodnotu 488 bps. Je určena jako dvojnásobek doby přetečení 8 bitového čítače/časovače 0 s předděličkou 8 a frekvencí hodinového signálu 2 MHz. Modulační rychlost je ve skutečnosti dvojnásobná, což je způsobeno použitím kódování Manchester.

4.1. Komunikační protokol

Návrh komunikačního protokolu vychází jednak z požadavků použitého hardware a zároveň z požadavků na jistou míru zabezpečení dat vůči chybám, které mohou vzniknout během průchodu signálu rádiovým prostředím. Jak již bylo výše zmí-

něno, vysílač i přijímač pracují se 100 % modulací AM. Tyto moduly proto potřebují vyvážený poměr jedniček a nul během vysílání, což je zajištěno použitím kódování Manchester. Kromě toho, že tímto kódováním vylepšíme poměr jedniček a nul, zavedeme i 50% redundanci a snížíme tak přenosovou rychlost na polovinu. Každý bit je zde totiž vyslán dvakrát. Nejprve je vyslán přímo a potom negovaně. Podobně by se dalo ještě použít bitové kódování 1/3 : 2/3. Zde je každý bit vyslán na třikrát. V první třetině se odešle nula, v druhé třetině konkrétní bit a ve třetí třetině odešleme jedničku. Kromě těchto a dalších podobných způsobů bitového kódování, lze vyvážený poměr jedniček a nul zajistit i pomocí bajtového kódování. Příkladem je kódování FEC, kde je poprvé bajt vyslán přímo a potom negovaně.



Obr. 16: Jeden paket vysílaných dat v kódování Manchester

Jak je na první pohled patrné z obrázku, jako první vysíláme 3 bajty synchronizace (0xff), za níž následuje bajt dat, který je pětkrát zopakován. Jednotlivé bajty jsou od sebe vzájemně odděleny pomocí 2 bitů. První z nich je trojnásobné délky a druhý je standardně dlouhý.

Tato pětinašobná redundance má své opodstatnění v rámci zabezpečení dat vůči chybám během přenosu. V přijímači se po přijetí kompletního datového paketu sečtou jedničky ve všech bajtech na odpovídající pozici a vyhodnotí se chyba. Pokud je ve výsledku na určité pozici 3 – 5 jedniček, je na této pozici vyhodnocena log 1. Pokud však na dané pozici napočítám 2 a méně jedniček, pozice bude ve výsledku obsazena nulou. Jistě, našla by se řada způsobů, jak data zabezpečit jinak a lépe využít pětinašobnou redundanci. Rozhodl jsem se pro použití tohoto jednoduchého způsobu z důvodu rychlého výpočtu na přijímací straně a možnosti odstranění hned několika chyb a to i na stejných pozicích v bajtu.

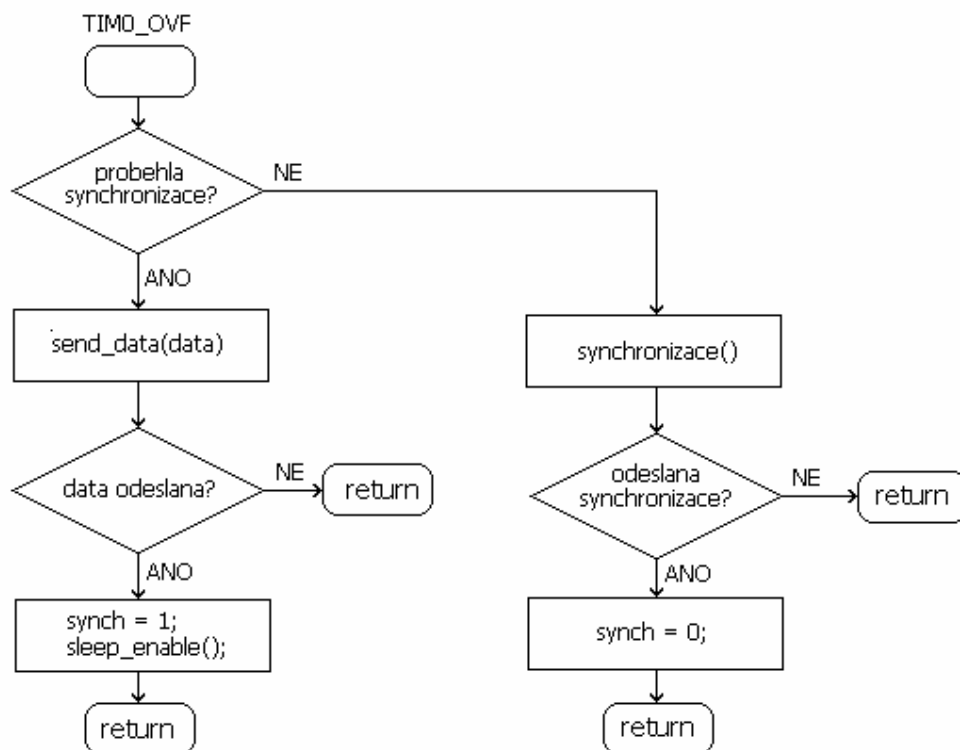
4.2. Realizace software vysílače

Celý zdrojový kód software vysílače je uveden v příloze na doprovodném CD. Software vysílače je napsán v prostředí AVR Studia v jazyce C. Byl testován a laděn přímo v zařízení, na desce vysílače bezdrátového příposlechu. Mikrokontrolér vysílače je taktován krystalem o kmitočtu 2 MHz a je využit režim Power down tak často, jak jen je to možné. Využity jsou celkem dva zdroje přerušení. Jedná se o externí přerušení aktivované náběžnou hranou signálu INTO a o přerušení při přetečení čítače/časovače 0. Prakticky celý program je prováděn během obsluhy těchto přerušení.

Po proběhlém resetu mikrokontroléru je zavolána funkce *setup()*, kde jsou nastaveny některé důležité bity v kontrolních registrech. Je povoleno externí přerušení, citlivé na náběžnou hranu. Dále jsou nastaveny parametry čítače/časovače 0, povo-

len úsporný mód a globálně povoleno přerušení. Na konci funkce *setup()* je procesor uspán (aktivace módu Power down).

Samotné vysílání probíhá následovně. Aktivací externího přerušení INT0 je procesor probuzen z Power down módu a začíná obsluha přerušení. Je načtena hodnota registru PINA a uložena do pomocné proměnné s tím, že bity 0 – 2 odpovídají vstupním kanálům a je třeba je posunout na pozice 5 – 7. Poté je hodnota pomocné proměnné sečtena se zvoleným identifikačním číslem sady. Získali jsme data určená k odeslání. V této fázi je třeba zakázat externí přerušení a aktivovat čítač/časovač 0. Ten je provozován v normal módu s předděličkou 8, při přetečení je aktivováno příslušné přerušení.



Obr. 17: Vývojový diagram obsluhy přerušení čítače/časovače 0

Prvním přerušením od č/č 0 je zahájeno samotné vysílání, podle vývojového diagramu viz obr. 17. Hned první podmínkou ověříme, zda máme odvyšlat trojici synchronizačních bajtů, nebo jestli už nevyšíláme bajty datové. V deklaraci proměnných je defaultně nastaven příznak vysílání synchronizace na log 1. To znamená, že se hned v první podmínce dostáváme do její else větve a je zahájeno odesílání synchronizace. Tento úkol je vyřešen během volání funkce *synchronizace()*, která vrací hodnotu log 0, dokud neproběhne synchronizace. V těle této funkce, je volána univerzální funkce pro odeslání jednoho bajtu *send_byte(char poslat)*. Tato funkce zajišťuje kódování Manchester a vrátí jedničku po odeslání celého bajtu. Po odeslání tří bajtů je volána funkce *start_stop_bit()*, která se postará o vygenerování start/stop bitu.

V okamžiku, kdy tato funkce vrátí log 1, je vynulován příznak vysílání synchronizace a funkce *synchronize()* vrátí též log 1. V příštím přerušení od čítače/časovače 0 je zahájeno samotné odesílání dat.

Data jsou odesílána během volání funkce *send_data(char data)*. Na začátku této funkce je testován příznak prvního datového bajtu kvůli nutnosti generovat start/stop bit, který je již obsažen na konci funkce *synchronize()*. Pro odeslání datového bajtu je opět volána universální funkce *send_byte(char poslat)*. Po odvysílání prvního bajtu je vynulován příznak prvního vysílání, je nastaven příznak potřeby dalšího oddělovacího start/stop bitu, také je inkrementováno počítadlo odeslaných datových bajtů a následuje konec obsluhy přerušení. Při jeho dalším příchodu musíme začít odesílat oddělovací start/stop bit.

Jakmile je tento úkon dokončen, nulujeme příznak oddělovače a ověříme, zda náhodou nebyl předchozí datový bajt poslední. Pokud byl poslední, nastavíme výchozí hodnoty všech důležitých příznaků a počítadel a vrátíme jedničku. Pokud však předchozí bajt poslední nebyl, funkce vrací nulu a v dalším přerušení začíná vysílání následujícího datového bajtu. Po jeho odeslání je opět inkrementováno počítadlo odeslaných bajtů a nastaven příznak oddělovače. V této fázi je třeba testovat, jestli právě odeslaný datový bajt nebyl pátý v pořadí. Pokud ne, pokračujeme standardním způsobem, ale pokud byl poslední, je nastaven příznak posledního bajtu, kvůli korektnímu ukončení vysílání. Jakmile jsou všechna data odeslána, krátce rozsvítíme indikační diodu LED na portu A, vypneme čítač/časovač 0, nastavíme příznak synchronizace, počkáme 150 ms, zhasneme LED, vypneme vysílač, povolíme přerušení INT0, aktivujeme mód Power down a čekáme na příchod dalšího externího přerušení, po kterém bude celý proces zopakován.

4.3. Popis jednotlivých funkcí programu vysílače

Funkce, které jsou volány během průběhu programu vysílače shrnuje následující výpis:

setup(void)

Tato funkce je volána pouze jednou a to na samém počátku programu. Má na starost korektní nastavení, případně aktivaci důležitých periférií, které budou následně využívány v průběhu programu.

synchronize(void)

První volání této funkce přichází s prvním přetečením čítače/časovače 0 a zajišťuje správné odvysílání synchronizačních bajtů. Její detailní popis se nachází v předchozí kapitole.

start_stop_bit(void)

Funkce mající na starost generování start/stop bitu, který je vyslán po průchodu synchronizace a pak za každým odvysílaným bajtem dat, včetně toho posledního. Start/stop bit je rozložen celkem do tří částí. První část je log 1 trvajícím trojnásobek délky standardní log 1 synchronizace nebo dat. Následuje druhá část a tou je trojná-

sobně dlouhá log 0. Start/stop bit je uzavřen odvysíláním standardní log 1 v kódu Manchester. Hned na počátku vysílání start/stop bitu je testováno počítadlo pozice (abychom věděli, v jaké části start / stop bitu se právě nacházíme). Pokud je počítadlo < 3 je na aktivní pin portu A odeslána log 1. Pokud je hodnota počítadla > 3, ale zároveň nižší než < 6, posíláme na pin portu A log 0. Pokud je však hodnota počítadla = 6 testujeme příznak log 1 nebo log 0 k odeslání závěrečného bitu. Odešleme-li log 0, nastavíme příznak na log 1 a vrátíme log 0. Během posledního volání programu je odeslána právě poslední log 0, jsou nastaveny výchozí hodnoty pro další použití a je vrácena log 1. Tím vysílání start/stop bitu končí.

char send_byte(char poslat)

Jedná se o universální funkci pro odeslání jednoho datového nebo synchronizačního bajtu, která zajišťuje mimo samotného odeslání bajtu především jeho zakódování kódem Manchester.

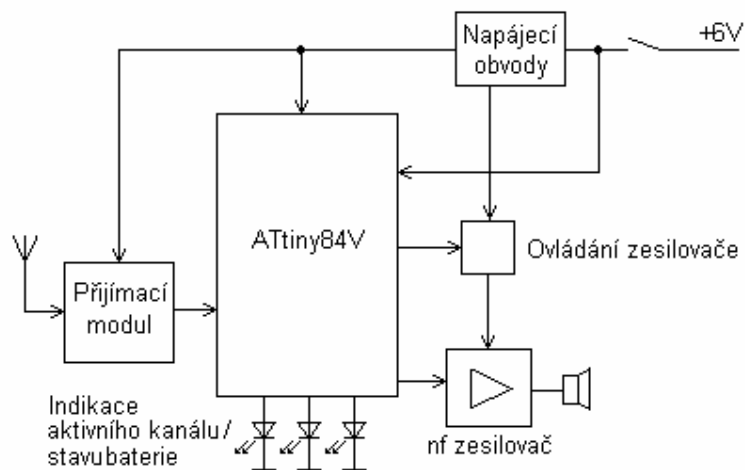
Po zavolání funkce jsou alokovány dvě pomocné, lokální proměnné, které jsou následně využity při práci s portem procesoru. Provedeme testování příznaku prvního volání funkce. Pokud je volána prvně, uloží vstupní datový bajt do dvou pomocných proměnných a to přímo a negovaně, což se provádí pomocí logické funkce XOR. Následuje změna příznaku prvního vysílání. Otestujeme příznak pozice bitu (jestli odesíláme přímou část nebo negovanou). V našem prvním volání funkce odesíláme nejprve přímou část bitu. Inkrementujeme počítadlo přímých „půlbitů“ a do lokální pomocné proměnné vložíme data k odeslání bitově posunutá. Posunujeme „aktivní“ bit na pozici pinu portu A užívanému k vysílání. Nakonec odešleme bit na port PORTA. Ještě je třeba otestovat, zda oba čítače jak přímých, tak negovaných bitů nedosáhly hodnoty 8. Pokud ne, nic neděláme, pouze vrátíme log 0. Pokud by dosáhly hodnoty 8, znamenalo by to, že byl odeslán celý datový bajt. V takovém případě se nastaví výchozí hodnoty všech použitých proměnných a příznaků a vrátíme log 1.

send_data(char data)

Toto je funkce, která plně využívá předchozí uvedenou funkci *send_byte* a mezi jednotlivé datové bajty vkládá oddělovací start/stop bity. Po odeslání 5 datových bajtů vrací funkce hodnotu 1. Detailní popis jejího chování se nachází v předchozí kapitole.

5. Obvodové řešení přijímače příposlechu

5.1. Blokové schéma přijímače

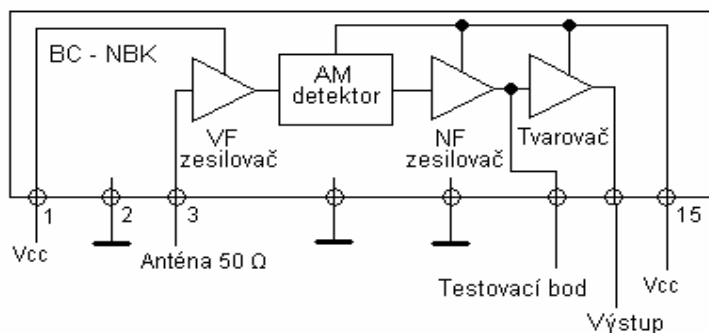


Obr. 18: Blokové schéma přijímače bezdrátového příposlechu

Na obr. 18 se nachází zjednodušené blokové schéma přijímací části sady bezdrátového příposlechu k elektronickým signalizátorům záběru. Všechny vyobrazené bloky budou detailně probrány v následujícím textu.

5.2. Vstupní obvody přijímače – použitý hybridní modul

Vysokofrekvenční vstupní obvody přijímače jsou vyřešeny pomocí hybridního modulu typu BC-NBK [5] firmy Aurel [1]. Tento modul nebyl vybrán zcela náhodně, ale na základě jeho velmi nízké ceny, nízké požadované přenosové rychlosti a v neposlední řadě i díky jeho dostupnosti. Navíc se jedná o přijímací modul přímo určený k použití s modulem vysílače TX-SAW/433 s-z [4] od stejného výrobce. Volba těchto dvou modulů se v konečném výsledku neukázala jako optimální řešení, o čemž se blíže zmíním v následujícím textu.



Obr. 19: Vnitřní blokové schéma modulu BC-NBK [7]

Z vnitřního blokového schématu na obr. 19 je patrné, že se jedná o přímozesilující přijímač amplitudově modulovaných signálů na kmitočtu 433,92 MHz. Tento typ přijímače nemá na svém vstupu zařazen žádný pásmový filtr, obsahuje pouze pevně naladěný, jednoduchý rezonanční obvod zařazený mezi VF zesilovač a detektor. VF zesilovač tvoří dva bipolární VF tranzistory. Tento zesilovač je následován superreakčním detektorem. Toto řešení s sebou nese kromě výhody vyšší citlivosti při malém počtu součástek i některé poměrně zásadní nevýhody. Největší nevýhodou je poměrně vysoká úroveň vlastního šumu, který je natolik silný, že dokáže občas překloupat komparátor, který zde plní funkci tvarovače signálu. Funkci tohoto tvarovače a následujícího NF zesilovače tvoří dvojitý operační zesilovač typu LM358.

Mimo typu BC-NBK dodává firma Aurel na trh ještě jinou variantu tohoto přijímače s označením NB-CE. Tento přijímač je téměř identický. Rozdíl je pouze ve vstupní části, kde má typ NB-CE zařazenu úzkou pásmovou propust, čímž se zvýší odolnost vůči rušení a celková selektivita avšak za cenu mírného snížení citlivosti.

Maximální kmitočet obdélníkového signálu na výstupu přijímače BC-NBK je 2 kHz. Je-li anténa bez signálu, na výstupu přijímače by se měla objevit stálá nízká úroveň. Přijímač BC-NBK je připraven pracovat asi 2 s po zapnutí napájecího napětí, modul NB-CE přibližně o 0,5 s později.

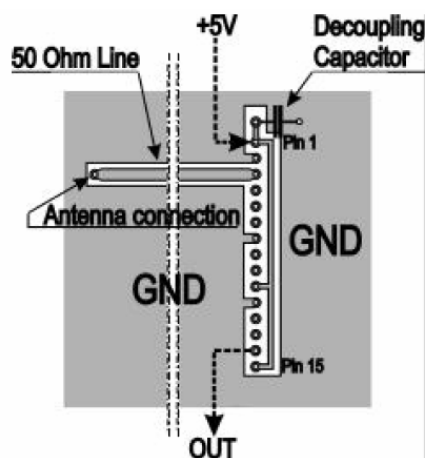
Tento modul vyžaduje vyvážený poměr jedniček a nul ve vysílaném signálu. Obvod tvarovače je spolu s NF zapojen tak, že na výstupu NF zesilovače je zapojen integrační člunek, který se nabíjí právě na střední hodnotu přijímaného signálu. Pokud tato hodnota výrazněji kolísá, dochází ke zkreslení časového průběhu přijímaného signálu vzhledem k signálu vysílanému. Proto je třeba před každým přenosem dat odvídat přibližně 100 ms dlouhý úsek symetrického signálu (101010101 apod.), přesně jak je naznačeno v popisu komunikačního protokolu. Od výrobce je zde nastavena jistá „šumová imunita“. Jedná se o překlápěcí úroveň komparátoru, jejíž tovární hodnotu okolo 100 mV, lze jen velmi obtížně upravit.

Jak bylo již uvedeno, tyto moduly pracují s dvoustavovou AM modulací (ASK, OOK). To s sebou nese další nevýhodu, která je podle mého názoru hodná zmínky. Na rozdíl od modulů s modulací např. 2 FSK se zde projevuje negativní vliv impulsního rušení amplitudového charakteru. Úder blesku, nebo nedbale odrušený přerušovač motoru motocyklu se projeví impulsem nebo jejich skupinou na výstupu tvarovače signálu.

Specifikace správného použití modulu

Modul BC-NBK má rozměry 38,1 x 13,7 x 5,5 mm. Rozsah napájecího napětí je 4,5 – 5,5 V. Napájecí napětí musí být stabilizované a pečlivě vyhlazené, s čímž nebudou v aplikaci napájené z baterií velké potíže. Tato podmínka zde je z důvodu nastavení překlápěcí prahové úrovně komparátoru, která není pevně stabilizována, ale je nastavena pouze relativně odporovým děličem. Při zvlnění napájecího napětí by se tato úroveň začala postupně měnit a tím by mohly vznikat nežádoucí impulsy na výstupu. Dále je nutné použít blokovací keramický kondenzátor o kapacitě 100 nF co nejbližše napájecím vývodům modulu. Výrobce důrazně doporučuje umístit modul na oboustrannou desku s plošnými spoji se souvislou zemnicí plochou po obou stranách.

Všechna tato opatření jsou kvůli omezení rušivých vlivů okolních elektronických součástek.



Obr. 20: Příklad vhodného umístění modulu přijímače BC -NBK na desku plošných spojů [5]

Modul je nutné umístit na desku plošných spojů pokud možno co nejdále od mikroprocesoru. Vzhledem k tomu, že se tu vyskytují obdélníkové signály, vznikají harmonické produkty o vysokých frekvencích. Od ostatních aktivních i pasivních součástek je doporučena minimální vzdálenost 5 mm z důvodu nebezpečí ztlumení rezonančního obvodu a případných nežádoucích vazeb. Anténní vedení o impedanci 50 Ω je třeba navrhnut co nejkratší a ve vzdálenosti nejméně 5 mm neumístit žádné součástky. Místo připojení antény (tj. konec vedení) musí mít kolem sebe 5 cm volného prostoru v případě použití prutové antény přímo na desce nebo 5 mm prostoru, použijeme-li koaxiální kabel. Tato poslední podmínka není u prototypu dodržena. Testování probíhalo pouze za „pokojových“ podmínek a koaxiální kabel s prutovou anténou na zvláštní destičce by překážel a způsoboval další těžkosti. Ve finální verzi produktu bude i tato podmínka pochopitelně důsledně dodržena. O vhodných typech antén použitelných pro tento modul a zároveň i pro modul vysílače pojednává kapitola 7.

5.3. Použitý mikroprocesor a jeho zapojení

Když jsem se rozhodoval, který typ mikroprocesoru použiji ve svém přijímači, bylo naprosto jasné, že se bude opět jednat o některý typ z produkce firmy Atmel [2] a pochopitelně to bude AVR. Bylo by krajně nešikovné použít v jednom zařízení dvě různé platformy mikrokontrolérů a to především z důvodu ladění aplikací.

Při výběru konkrétního typu mikrokontroléru mě zajímaly zejména parametry, jako je spotřeba v aktivním módu a samozřejmě i v režimu spánku, dále byla důležitá velikost paměti FLASH, protože software přijímače je výrazně rozsáhlejší než vysílače. V neposlední řadě byl podstatný také počet vstupně-výstupních bran, kterými obvod disponuje. Mezi typy, které jsem zařadil do užšího výběru, byly ATmega8, ATtiny26L a ATtiny84V.

ATmega8 obsahuje 8 kB paměti FLASH a 1 kB paměti SRAM, v aktivním módu dosahuje při napájecím napětí 3 V a taktovacím kmitočtu 4 MHz proudového odběru 3,6 mA, v režimu spánku je odběr proudu asi 500 nA. Mikroprocesor je dodáván v pouzdrech DIP 28, TQFP 28 a MLF 28.

ATtiny26L má integrovány 2 kB paměti FLASH a 128 B paměti SRAM. V aktivním módu je jeho spotřeba 0,7 mA při napájecím napětí 3 V a taktovacím kmitočtu 1 MHz. V režimu spánku obvod odebírá necelou 1 μ A. Tento procesor je dodáván v pouzdrech DIP 20, SOIC 20 a MLF 20.

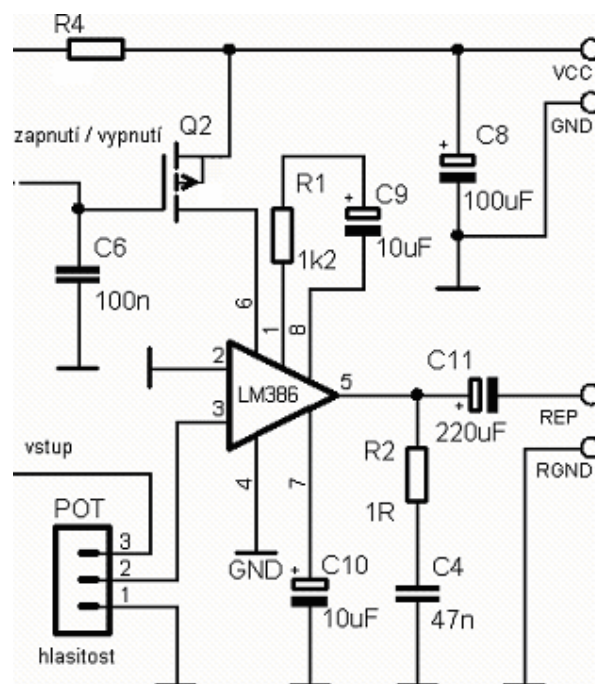
Mikroprocesor ATtiny84V byl již popsán v kap. 3.5. Pro tento typ jsem se nakonec rozhodl. Tato volba byla podmíněna především jeho spotřebou v obou výše uvedených provozních režimech, jeho rozměry, počtem použitelných vývodů a integrovaných periférií. Další věc, která svědčila ve prospěch tohoto obvodu je fakt, že se jedná o jeden z nejnovějších typů, nabízených firmou Atmel.

Obvod je zapojen zcela standardně dle doporučení výrobce. Resetovací obvod je tvořen kondenzátorem C3 a rezistorem R3. Jeho časová konstanta je i v přijímači zvolena 0,1 ms. Taktovací kmitočet hodinového oscilátoru jsem nakonec zvolil poněkud vyšší, než jsem měl původně v plánu. Jeho hodnota 2 MHz je zcela dostačující pro běh programu přijímače a vyhodnocování přijímaného signálu je zabezpečeno s dostatečnou rezervou. Samotné naprogramování mikroprocesoru je opět zajištěno pomocí sériového programátoru ISP. V zařízení je k tomuto účelu určen standardní konektor ISP.

5.4. Nízkofrekvenční zesilovač

Primárním úkolem přijímače a vlastně celého příposlechu je vyvolání včasného a zejména hlasitého poplachu během záběru ryby. Hlasitost musí být dostatečná i k probuzení spícího člověka. Z těchto požadavků jasně plyne nutnost použití reproduktoru.

Vzhledem k tomuto faktu, je třeba výstupní akustický signál řádně výkonově zesílit. K tomuto účelu jsem si vybral známý integrovaný výkonový zesilovač LM386 [8]. Obvod dokáže dodat výkon až 0,5 W do zátěže 4 nebo 8 Ω . Lze u něj též nastavit celkové napěťové zesílení v rozmezí 20 – 200 pomocí několika málo externích součástek. Vyznačuje se mimo jiné i poměrně nízkou spotřebou a širokým rozsahem napájecího napětí 4 – 15 V. Na druhou stranu obvod je nechvalně znám vyšším vlastním šumem, což však v dané aplikaci nevádí. Obvod LM386 je dodáván v pouzdře DIP 8.



Obr. 21: Schéma zapojení NF zesilovače

Na obr. 21 je schéma zapojení koncového NF zesilovače. Tranzistor Q2 zde slouží k ovládání napájení zesilovače. Dokud nebudou korektně přijata a vyhodnocena data o poplachu, je zesilovač odpojen od napájení. V případě generování poplašného signálu se konkrétní větev programu postará o jeho neprodlené zapnutí. Stejně jako v obvodu vysílače je i zde použit tranzistor MOSFET s indukovaným kanálem P typu BS250.

Na první pohled je patrné, že se jedná o katalogové zapojení tohoto integrovaného obvodu. Na vstup je zařazen potenciometr regulace hlasitosti, který je připojen ke konektoru POT. Nebude umístěn přímo na desce plošných spojů, ale bude vyveden na vhodné místo přístrojové skříňky.

5.5. Napájení přijímače bezdrátového příposlechu

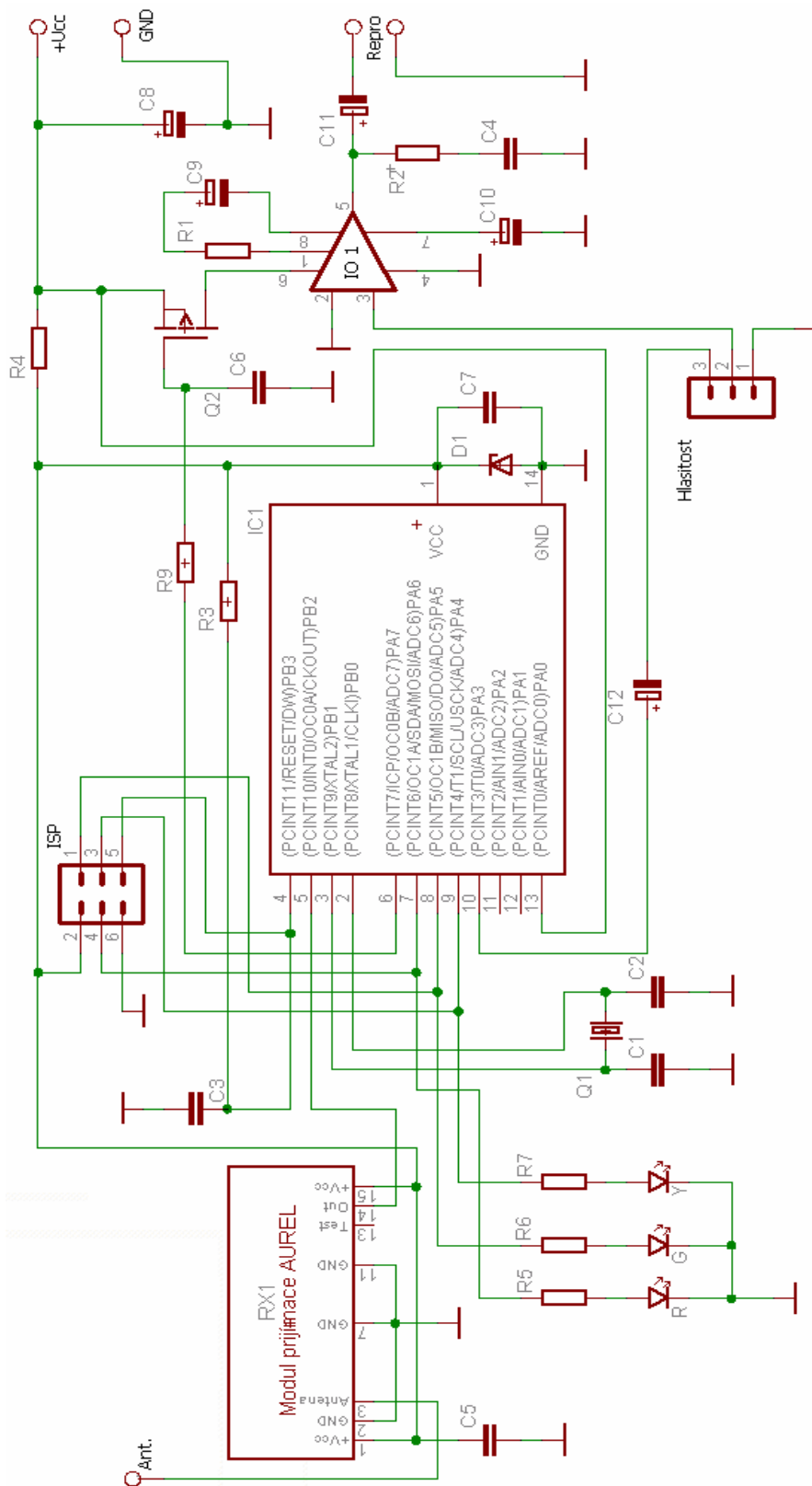
Jak bylo již výše uvedeno, rozhodl jsem se přijímač napájet ze čtyř článků AA. Výkonový zesilovač je napájen přímo z těchto baterií, jelikož nepotřebuje pro svou funkci stabilní napájecí napětí. Pro napájení zbytku zařízení včetně mikroprocesoru a modulu přijímače je napětí stabilizováno na hodnotu 4,2 V pomocí parametrického stabilizátoru se Zenerovou diodou. Stejně jako v případě vysílače vím, že se nejedná o zcela optimální řešení a to z důvodu vyššího klidového proudu diodou a také kvůli nižšímu činiteli stabilizace. Vhodnější se jeví použití obvodů jako jsou TL431 nebo LM317L a řady dalších.

5.6. Schéma zapojení přijímače

Na obr. 22 je vyobrazeno kompletní schéma zapojení přijímací části sady bezdrátového příposlechu k elektronickým signalizátorům záběru. Jednotlivé funkční blo-

ky obvodu byly rozebrány v předešlém textu. Jediné, co ještě nebylo zmíněno je trojice signalizačních diod LED připojených k portu mikrokontroléru. Tyto LED diody slouží k optické indikaci kanálu během signalizace poplachu. Současné verzi software přijímače nejsou tyto diody využity k žádnému jinému účelu. Přímo se však nabízí možnost testování stavu baterií při každém zapnutí přijímače. Použitý mikrokontrolér totiž disponuje integrovaným AD převodníkem, který obsahuje i vnitřní zdroj referenčního napětí.

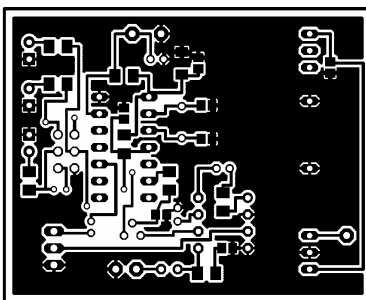
Stejně jako v případě vysílače bylo i schéma zapojení přijímače a desky plošných spojů vytvořeno ve vývojovém prostředí Eagle firmy CADSoft [6]. Seznam součástí přijímače sady bezdrátového příposlechu je obsažen v příloze na konci dokumentu.



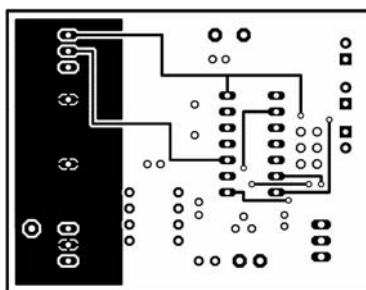
Obr. 22: Celkové schéma zapojení přijímače bezdrátového příposlechu

5.7. Deska s plošnými spoji přijímače

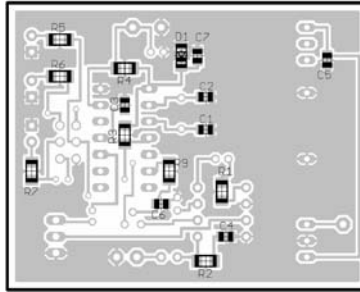
Deska s plošnými spoji přijímací části je vyrobena z oboustranně plátovaného kuprexitu se souvislou zemnicí plochou po obou stranách. Její rozměry jsou 44,5 x 56,5 mm. Na desce jsou umístěny všechny součástky přijímače kromě potenciometru ovládání hlasitosti, vypínače a reproduktoru.



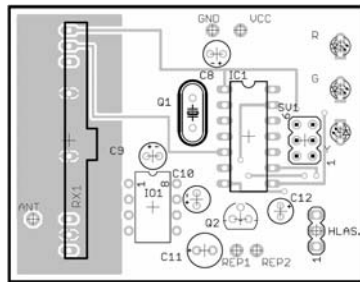
Obr. 23: Deska s plošnými spoji přijímače – strana spojů



Obr. 24: Deska s plošnými spoji přijímače – strana součástek



Obr. 25: Rozmístění součástek přijímače – strana součástek



Obr. 26: Rozmístění součástek přijímače – strana spojů

6. Obslužný software přijímače

Nejprve je nutné stanovit požadavky, které klademe na použitý mikrokontrolér a program v něm běžící. Jde zejména o nalezení kompromisu ve vhodném využití výpočetního času mikrokontroléru. Zjednodušeně řečeno, přímo v době kdy přijímáme data nemůžeme s hardware tak jak je navržen, provádět signalizaci.

Ve vysílači je za posledním odeslaným bitem vložena 100 ms dlouhá pomlka, která zajišťuje jistou minimální dobu, kdy bude signalizace zaručena. Těchto 100 ms nebudou vyhodnocována žádná data. Samotné vyhodnocení dat trvá přibližně 360 ms. Fakt, že se data vytváří a odesílají jen po příchodu aktivní hrany na vstup INT0 ve vysílači a minimální doba signalizace poplachu i při jednotlivém impulsu, zajišťují určitou jistotu vykonání dostatečné signalizace.

Přijímač je od počátku navrhován, jako poměrně jednoduché zařízení, obsahující pouze dvě zásadní součástky a to mikrokontrolér a hybridní modul BC-NBK. Otázku problémů spojených s výslednou signalizací jsem příliš neřešil až do okamžiku, kdy jsem měl takřka hotový software přijímače. Zjistil jsem, že by bylo ideální použít další mikrokontrolér (některý z nejmenších a nejjednodušších typů) nebo nějaký jednoúčelový obvod, který by zajišťoval pouze optickou a akustickou signalizaci poplachu. Vzhledem k minimu použitých externích součástek a použití interního RC oscilátoru by toto řešení bylo i poměrně levné. Hlavní mikrokontrolér by pouze předal informaci o aktivaci / deaktivaci poplachu a o kanálu, na kterém k tomu došlo. Všechny ostatní úkony spojené s generováním akustického a optických signálů by zajišťoval druhý mikrokontrolér.

6.1. Koncepte software přijímače

Výstupní pin hybridního modulu přijímače BC-NBK je připojen ke vstupu externího přerušení INT0, jehož citlivost je nastavena na náběžnou hranu přicházejícího signálu. Po dobu neaktivity je z důvodu úspory energie procesor v módu Power down a z toho je následně probouzen příchodem náběžné hrany na vstup INT0 a vyvoláním externího přerušení. V tuto chvíli je zahájen příjem synchronizace, případně dat, pokud synchronizace proběhla.

Samotný příjem a vyhodnocení dat je založeno na čtení stavu 16bitového čítače/časovače 1 v okamžicích příchodů externích přerušení. Čítač je nastaven do režimu „normal“ a je mu předřazena dělička 8, ta nám zajistí, že nepřeteče během příjmu synchronizace a zároveň máme možnost pracovat s přiměřeně vysokými hodnotami v registru TCNT1, tudíž přesnost bude také vyhovující.

Pokud prozatím vynechám snahu o korekci chyb způsobených rušením, proces dekódování dat vypadá zhruba následovně. Je důležité připomenout, že přichodící data jsou v kódování manchester a tomu musí odpovídat i způsob jejich čtení. Použitá metoda čtení dat je založena na znalosti předchozího korektně přijatého bitu a hodnoty TCNT1, která byla dosažena v čase mezi dvěma náběžnými hranami přichodících dat.

Jednotlivé bajty dat jsou od sebe odděleny start/stop bity stejně, jako je blok synchronizace oddělen od prvního bajtu dat. Proto musí být tento start bit zahrnut do dekodovací tabulky a musím s ním v programu počítat, vždy ho správně vyhodnotit, nepočítat za data, ale pouze posunout ukazatel v datech na další bajt. Rozdíl současného a předchozího stavu čítače potom porovnávám s jednotlivými pozicemi v tabulce přepočtených hodnot a pokud je v pořádku tato hodnota a zároveň předchozí bit, uložím přijatý bit na svoji pozici do aktuálního bajtu v poli přijatých dat.

Tímto způsobem získáme 5 bajtů dat. Nyní je potřeba využít této pětinasobné redundance a vypočítat korektní data. Algoritmus výpočtu je následující: Na pozici 0 – 7 sečteme počet log 1 ve všech bajtech a tam, kde jich je 3 a více, vyhodnotíme jako výsledek log 1. Pokud jich však někde bude 2 a méně, výsledkem je pochopitelně log 0. Je zcela jasné, že tato poměrně jednoduchá korekce rozhodně nedokáže opravit shluky chyb, ale jen jednotlivé náhodné chyby. Celkem tedy dokážeme eliminovat 2 chyby na stejné pozici v každém bajtu, což je dohromady 10 chyb v celém přenosu. Důvodem použití této metody je fakt, že si při nízkém taktovacím kmitočtu nemohu dovolit příliš složité výpočty, neboť by byly časově náročné, tím myslím náročnější než tato metoda. Pro popsané řešení mluví i fakt, že se neprovádí žádný „postprocessing“, ale všechny výpočty se provádí v reálném čase.

Provedením korekce dat končí samotný příjem. Následuje porovnání prvních 5 bitů s identifikačním číslem přijímače a za předpokladu, že během přenosu nenastaly chyby, je spuštěn poplach (generování akustického a optického signálu zajišťuje k tomu určená funkce ve smyčce hlavního programu). Po provedení 2,5 s trvajících poplachu bude mikrokontrolér uveden do režimu Power down.

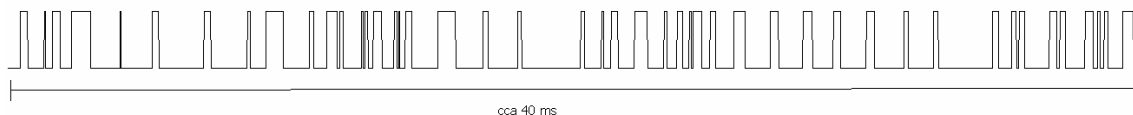
6.2. Chování software přijímače v reálné situaci

Následuje popis několik poznatků nutných, ke správnému pochopení činnosti software přijímače, jako celku, ale i k pochopení činnosti částí některých funkcí, použitých během příjmu.

Teoreticky by mělo být napsání a odladění celého software přijímače snadnou záležitostí, což se v praxi zásadně nepotvrdilo. Výrobce totiž ve svých materiálech k modulům BC-NBK uvádí na výstupu nízkou úroveň, pokud je anténa bez signálu. Z tohoto tvrzení jsem při návrhu vycházel, ale nakonec se ukázalo, že není zcela pravdivé. Dopustil jsem se chyby a neujistil se, co se skutečně děje na výstupu modulu bez signálu.

Jak jsem zmínil v kapitole 5.2. , modul přijímače obsahuje superreakční detektor, který se vyznačuje především svým vysokým vlastním šumem. Tento šum dosahuje naprázdno takových hodnot, že dochází ve velmi nepravidelných intervalech k překlápění komparátoru tvarovače výstupního signálu a na výstupu se tudíž objevují shluky impulsů o různé délce, které jsou někdy i poměrně pravidelné. Za situace, kdy je výstup přijímače připojen přímo na vstup externího přerušení, které má zároveň sloužit i k aktivaci mikrokontroléru z režimu spánku, je zcela vyloučeno použití tohoto režimu. Tím pádem se stává praktická použitelnost tohoto přijímače značně diskutabilní. Kromě toho, že přijdeme o úsporu energie díky uspanému procesoru,

i modul přijímače vykazuje zvýšený odběr, protože je po celou dobu aktivní a zpracovává „data“. Nárůst spotřeby může dosáhnout takřka 10 mA. Tuto velmi nepříjemnou vlastnost modulu BC-NBK dobře zachycují následující obrázky sejmuté logickým analyzátozem přímo z výstupu modulu.



Obr. 27: Šum pronikající na výstup přijímače

Na obr. 27 je zachycen příklad, jak může vypadat šum pronikající na výstup modulu přijímače v klidovém stavu. Délka vyobrazeného úseku je sice přibližně 40 ms, ale množství a „frekvence“ tohoto šumu se v průběhu času ztelně mění. Tyto změny zachycuje následující obrázek.



Obr. 28: Jiná varianta šumu objevujícího se na výstupu

Obr. 28 zachycuje zcela odlišnou situaci na výstupu hybridního modulu přijímače v klidovém stavu. Chování superreakčního detektoru ve stavu bez signálu ovlivňuje řada faktorů, jako je například přítomnost zapnutých elektrospotřebičů (počítač, televize atd.).

Z výše uvedených skutečností je naprosto jasné, že je potřeba eliminovat (ignorovat) tyto rušivé impulsy, což je navíc znesnadněno i tím, že jsou často poměrně pravidelné. Další drobná vada je v samotném vysílači. Přestože jsou jednotlivé poloviny bitů včetně synchronizace generovány pomocí stejných funkcí, dochází zde díky větvení programu k tomu, že délky trvání vysílaných polovin bitů nejsou přesně stejné. Jedním z možných řešení by bylo zařazení přesně vypočtených zpoždění, které by vyrovnaly doby průchodů kratšími větvemi programu s dobou průchodu tou nejdelší. Já se však rozhodl pro jiné řešení a to na straně přijímače. Zavedl jsem systém tolerance délek bitů, kdy je ještě hodnota napočítaná čítačem mezi předchozím a aktuálním přerušením (tj. 1 bit) považována za přijatelnou a tudíž i daný bit je korektní. Toto samozřejmě platí jen v případě příjmu synchronizace. Co se týče příjmu samotných dat a start/stop bitů, tolerance je nastavena na stejnou hodnotu, která bude upřesněna v popisu funkce konkrétních funkcí.

6.3. Popis funkce software přijímače

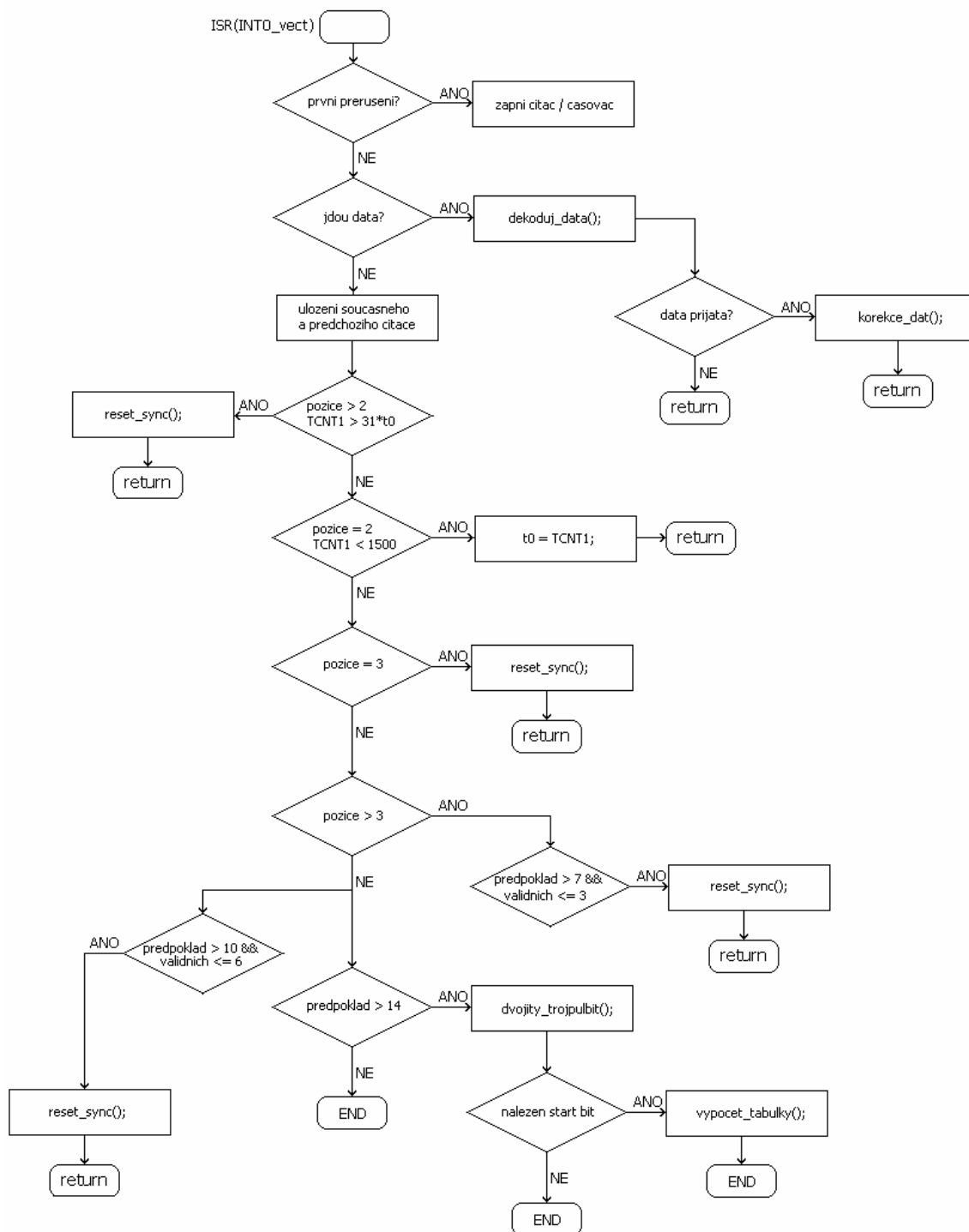
Celý zdrojový kód software přijímače je uveden v příloze na doprovodném CD. Nemyslím si, že by bylo vhodné, uvádět ho v plném rozsahu. V následujícím textu je detailně popsán.

Vzhledem k rušení, které produkuje sám přijímač, zde není použit úsporný mód Power down. Navíc je důležité zajistit, aby tyto impulsy nebyly pokud možno vyhodnoceny ani jako data, ani jako synchronizace. Samotný program bych rozdělil na dvě části. První část zajišťuje vyhodnocení synchronizace, příjem dat a jejich vyhodnocení v případě, že je vše v pořádku a nedošlo k chybám během přenosu (například ztráta start bitu nebo výpadek celého bajtu dat v důsledku rušení). Druhá je část funkcí, které nemají nic společného se samotným přenosem, nicméně se starají právě o ošetření případných chyb. Software přijímače funguje následujícím způsobem.

Po připojení napájecího napětí je proveden reset mikrokontroléru, jsou alokovány globální proměnné a následně je volána funkce *setup()*, kde probíhá nastavení použitých periférií. Je povoleno externí přerušení s aktivací náběžnou hranou vstupního signálu, samozřejmě je také nastaven 16bitový čítač/časovač 1 s předděličkou 8, která při hodinovém kmitočtu 2 MHz zajistí, že nedojde k přetečení čítače v době, kdy je přijímána synchronizace. Toto nastavení je poměrně hraniční, protože už malá změna hodinového kmitočtu (cca o 250 kHz) způsobila, že docházelo k přetečení čítače během příjmu posledního bajtu synchronizace. Samozřejmě jsou také příslušné piny mikrokontroléru nastaveny jako výstupní. Jedná se o ovládání LED diod, ovládání zesilovače a výstup akustického signálu.

Jakmile proběhne počáteční inicializace, je v těle hlavního programu otestován příznak vyvolání poplachu v minimální délce trvání 2,5 sec. Pokud není příznak nastaven, vstoupí program do nekonečné smyčky v těle hlavního programu, kde čekáme na příchod prvního přerušení.

V původní verzi software, kde jsem nepočítal s rušením od modulu přijímače, by zde nekonečná smyčka nebyla, ale program by po vyhodnocení příznaku poplachu aktivoval režim Power down a mikrokontrolér by čekal na aktivaci příchozím přerušením. Je sice možné režim Power down použít, ale nemyslím si, že je to přínosné, neboť by procesor přecházel mezi aktivním a uspaným stavem opravdu velice často, viz například obr. 27.



Obr. 29: Vývojový diagram obsluhy přerušení INTO

Obsluha externího přerušení INTO funguje v souladu s obr. 29. Příchodem prvního přerušení a je jedno zda se jedná o signál vysílače nebo o rušení z modulu BC-NBK, je otestována podmínka nultého přerušení. Pokud je splněna, program aktivuje čítač/časovač 1, inkrementuje počítadlo příchozích přerušení a ukončí obsluhu (žádné další podmínky nejsou splněny). V okamžiku příchodu další hrany uloží hodnotu stavového registru čítače TCNT1 do druhého prvku příslušného pole s tím, že předchozí

hodnotu (v tomto okamžiku blízkou nule) přesune do prvního prvku. Z těchto hodnot lze snadno vypočítat délku bitu (vzdálenost dvou náběžných hran na časové ose). Dále je inkrementováno počítadlo přerušení a v následné podmínce je otestováno, jestli není délka prvního přijatého prvku příliš velká. Tolerance je zde poměrně vysoká, jedná se o 1/3 předpokládané délky 1 bitu synchronizace (tj. TCNT1 = 1000). Pokud hodnota není v toleranci, je zavolána funkce *reset_sync()*, která nastaví výchozí hodnoty a začínáme opět od nuly. Pokud hodnota TCNT1 v toleranci naopak je, uložíme délku prvního bitu do proměnné referenční délky 1 bitu a ukončíme obsluhu tohoto přerušení. V následujícím 3. přerušení opět uložíme aktuální stav čítače, posuneme minulý a vypočteme délku aktuálního bitu. Také je otestováno, jestli není stav TCNT1 delší než předpokládaná délka celé synchronizace (plus tolerance 5 bitů). Pokud je délka překročena, program volá funkci *reset_sync()*. Jinak obsluha pokračuje.

V tomto okamžiku je prvně volána funkce *zpracovani_synchronizace(unsigned short int casovac)* a pokud vrátí 0, je proveden reset, protože první 2 bity nebyly stejně dlouhé a tudíž se nejedná o synchronizaci. Poté je testována podmínka, kdy musí být aspoň 3 přijaté bity z prvních 7 validní (tj. v toleranci vypočtené ve funkci *zpracovani_synchronizace*). Následuje podmínka, testující, jestli je aspoň 6 bitů z prvních 10 validních. Pro jistotu již od 14. přijatého bitu (připouštíme ztrátu poloviny synchronizace) začínáme vyhledávat stop bit synchronizace, který je zároveň start bitem dat.

Samotné vyhledávání zajišťuje funkce *dvojity_trojpulbit()*. Pokud je stop bit nalezen, je zavolána funkce zajišťující výpočet referenční tabulky pro dekódování dat z Manchesteru. Nakonec je nastaven příznak příchodu dat, která začneme číst hned, během obsluhy následujícího přerušení.

Obsluha každého přerušení během čtení dat začíná uložením aktuálního stavu TCNT1 a jeho vynulováním. Následně je volána nejrozsáhlejší a nejkompexnější funkce celého programu *dekoduj_data()*. Jakmile jsou všechna data přijata a je nastaven příznak jejich přijetí, voláme funkci *korekce_dat()*, jejímž provedením končí příjem dat. Vypneme čítač, vypneme externí přerušení a nastavíme výchozí hodnoty všech použitých proměnných a polí. Následuje ověření shody přijatých dat s identifikačním číslem přijímače. V tuto chvíli je nastaven příznak poplachu a data jsou předána funkci zajišťující signalizaci.

6.4. Popis jednotlivých funkcí programu přijímače

Funkce, které jsou volány během průběhu programu vysílače shrnuje následující výpis:

void setup(void)

Tato funkce je volána pouze jednou a to na samém počátku programu. Má na starost korektní nastavení a případně aktivaci důležitých periférií, které budou později využívány v průběhu programu.

zpracovani_synchronizace(unsigned short int casovac)

Toto je funkce, jejímž účelem je zpřesnit referenční délku jednoho bitu t_0 , kterou později použijeme pro výpočet dekódovací tabulky a dále probíhá počítání validních bitů synchronizace. Tento údaj je použit v podmínkách v těle obsluhy externího přerušení INT0.

Počínaje příjmem třetího bitu synchronizace, je tato funkce volána i v každém dalším přerušení. Jejím zavoláním je předán jako vstupní parametr aktuální stav čítače TCNT1, jsou alokovány pomocné lokální proměnné a ze současného t_0 je vypočtena hodnota 5 % tolerance. Navíc je z hodnoty čítače TCNT1, aktuálního t_0 a jeho tolerance vypočten předpokládaný počet bitů, které by již měly být přijaty. Tato hodnota je uložena v proměnné předpoklad_bitu a je použita v podmínkách v těle obsluhy INT0. Pomocí násobiče odchyly (který je uložen do zvláštní proměnné, kvůli přehlednosti) určíme maximální přípustnou odchylku, která bude od 3. bitu dvojnásobná, tj. 10 %. Provedeme výpočet teoretické hodnoty, kterou by měl obsahovat registr TCNT1 a porovnáme ji v následující podmínce s reálnou hodnotou TCNT1 (samozřejmě v rámci vypočtených tolerancí). Pokud hodnota TCNT1 leží v mezích tolerancí, je dané přerušení vyhodnoceno jako validní, je vypočtena nová a přesnější hodnota t_0 , inkrementováno počítadlo skutečně přijatých bitů a vrácena hodnota 1.

reset_sync(void)

Poměrně užitečná funkce, která je volána pokaždé, když není splněna některá z podmínek během příjmu synchronizace. Jejím účelem je opětovné nastavení některých proměnných, počítadel a příznaků do výchozích hodnot. Jedná se například o vynulování všech použitých počítadel a registru TCNT1.

dvojity_trojpulbit(void)

Od 14. přijatého bitu synchronizace je volána tato funkce, jejímž účelem je rozpoznat start bit dat. Jeho rozpoznání je řešeno tímto způsobem z důvodu vyšší odolnosti vůči chybám (výpadkům) během přenosu synchronizačních bitů. Nejjednodušší, avšak nejméně bezpečné řešení by bylo pozici start bitu vypočítat z počtu přijatých synchronizačních bitů. V tomto případě je však rozpoznání provedeno následujícím způsobem. Po zavolání funkce je znovu vypočtena hodnota jisté tolerance, do jejíž mezí se musí start bit vejít. Po té je porovnávána velikost posledního přijatého bitu s teoretickou hodnotou, kterou by měl dosahovat včetně tolerancí. Tato hodnota je vypočtena jako trojnásobek t_0 určeného během předchozího volání *funkce zpracovani_synchronizace(unsigned short int casovac)*. Pokud hodnota uložená v poli `minuly_bit[1]` spadá do rozsahu tolerance, znamená to, že se jedná o start bit dat a tudíž víme, že budou přijímána data. Následuje úprava hodnot některých proměnných, protože start bit nechceme počítat za datový bit. Je nastavena výchozí hodnota registru TCNT1, nastaven příznak příchozích dat, minulý bit je nastaven na 1 a funkce vrátí 1.

vypocet_tabulky(void)

Funkci pro výpočet referenční tabulky voláme pouze jednou a to v případě nalezení start bitu dat. Pomocí této tabulky jsou později přímo dekódována data. Bylo by

výpočetně zbytečně náročné provádět další operace se 16bitovými čísly při každém příchozím datovém bitu.

Ve dvojrozměrném poli `tabulka[8][4]` je uloženo 8 skupin hodnot a to předchozí bit, násobek času t_0 , dekodovaná data a délka dat (v některých případech jsou dekodovány 2 bity najednou). V průběhu této funkce jsou z hodnot násobků t_0 , uložených ve výše uvedené tabulce a skutečné hodnoty t_0 (včetně odchylek) vypočteny hraniční body intervalů, do kterých musí spadat právě vyhodnocovaný bit. Tyto vypočtené hraniční body jsou uloženy v poli `vypoctena_tabulka[8][2]`.

dekoduj_data(void)

Zajištění samotného dekodování přijatých dat z kódu Manchester je zajištěno právě v průběhu této funkce. Jedná se o nejdůležitější a zároveň nejrozsáhlejší funkci celého programu.

Po provedení inicializace pomocných lokálních proměnných je přepočtena délka posledního zachyceného bitu do rozměru, odpovídajícímu hodnotám v poli `vypoctena_tabulka[8][2]`. Všechny tyto hodnoty jsou čtyřikrát větší, než ve skutečnosti. Chtěl jsem se takto vyhnout použití desetinných čísel.

Nalezení konkrétního řádku v referenční tabulce a tím pádem správné vyhodnocení přijatých dat, se provádí v průběhu cyklu `for`. Jako první se vyhledává shodný minulý bit. Pokud není nalezen, cyklus pokračuje dalším průchodem. Toto se opakuje do doby jeho nalezení. Jakmile je nalezen konkrétní řádek, je nastaven příznak nalezení řádku, dále je uložen počet bitů, které jsme v tomto kroku detekovali (1 nebo 2) a detekované bity jsou uloženy. V tuto chvíli je potřeba rozhodnout, jaká data a jestli vůbec budu ukládat. Jedná se o situaci konce start bitu, může nastat stav, kdy přijímaná data začínají log 0 a tudíž dojde ke spojení konce start bitu s prvním bitem dat (v našem případě s výše uvedenou log 0). Tato situace je zde ošetřena zvlášť, mimo tabulku, stejně jako nakládání se samotným koncem start bitu, ikdyž za ním následuje log 1.

V tomto okamžiku ověříme, jestli jsme náhodou nepřijali již všechna data. Pokud ano, vynulujeme příznak čtení dat a nastavíme příznak jejich kompletního přijetí, provedeme reset a vrátíme se do obsluhy přerušování. Pokud však ještě nebyla všechna data přijata, uložíme právě přijatý bit do pole `dat` a posuneme se v něm o potřebný počet pozic (1 nebo 2). Dále musíme uložit příznak minulého bitu pro potřeby příjmu příštího bitu. Po inkrementaci počítadla bitů, ověříme, zda právě neprobíhá příjem start bitu oddělujícího další bajt a v případě že ano, opravíme hodnotu počítadla bitů a nastavíme minulý bit na 1. Nyní je vhodné ověřit, zda tento start/stop bit neuzavírá celý paket. Pokud ano, nebudeme se jím dále zabývat a provedeme akci ukončení příjmu (viz výše). Tímto prováděním funkce končí a program se posouvá na další krok.

Činnost uvedená v předchozím odstavci je provedena pouze za předpokladu, že během příjmu nedošlo k chybám a data byla přijata všechna. K tomu však nemusí dojít a je třeba takovou situaci adekvátně vyřešit. O to se stará další část této funkce. V případě, že není nalezen řádek v tabulce, což vždy značí nějakou chybu během

přenosu, provedeme některé další operace. Ověříme, zda není délka přijatého nevalidního přerušení menší, než $t_0/2$. Pokud je menší, přerušení zahodíme – jedná se o rušení. Pokud délka chybového impulsu velikostí odpovídá $t_0 / 2$ včetně tolerance, postupujeme podle toho jaký byl předchozí bit a pokusíme se zajistit, aby byl další bit uložen na místo, kam patří, případně (příliš dlouhá doba k dalšímu přerušení $> 7 * t_0$) ho zahodíme. Je ošetřen i případ poškození nebo ztráty oddělovacího start bitu. Místa nevalidních bitů jsou obsazena nulami a naší snahou je posunout ukazatel pozice v datovém bajtu na tak, aby byl další validní bit již umístěn správně. Pokud je těchto posunutí příliš velké množství, zahodíme celý paket a provedeme reset a nastavení výchozích hodnot použitých proměnných. Závěrem bych rád dodal, že tato část týkající se opravy chyb není zcela odladěna a testována. Nemusí být proto plně funkční.

korekce_dat(void)

Voláním této funkce zajistíme korekci chyb vzniklých během přenosu s tím, že paket nebyl předchozí funkcí vyhodnocen jako vadný a ignorován. Korekce je provedena tak, že počítáme počet log 1 na jednotlivých shodných pozicích každého přijatého bajtu. Pokud napočítám 3 a více jedniček, je daná pozice ve výsledném datovém bajtu vyhodnocena jako log 1, pokud však napočítáme 2 a méně jedniček je pozice vyhodnocena jako log 0.

signalizace(void)

Funkce zajišťující akustickou a optickou signalizaci vyvolaného poplachu je volána v těle hlavního programu při splnění podmínce korektního vyhodnocení přijatých dat. To znamená, že známe kanál(y), kde bude opticky signalizován poplach a zároveň se shodují identifikační čísla vysílače i přijímače. Zde, jako v jediné funkci jsou použity komponenty z knihovny delay.h. Generujeme přerušovaný akustický signál o kmitočtu necelé 3 kHz s frekvencí přerušování 40 Hz.

6.5. Ladění a testování software přijímače

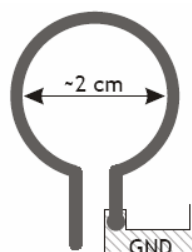
Vzhledem k většímu rozsahu a složitosti software přijímače, jsem jej většinu času ladil a testoval v mikrokontroléru ATmega16 v kombinaci s vývojovou deskou od firmy Mikroelektronika [10], kde bylo možno na LED diodách průběžně zobrazovat stavy vybraných proměnných a také bylo možno připojit jednoduchý logický analyzátor [11], který mi práci značně usnadnil. Během testování jsem si vytvořil řadu pomocných proměnných a ty odesílal na vybraný port mikrokontroléru, takže se mi nakonec podařilo zobrazit na logickém analyzátoru i jednotlivé průchody tabulkou pro dekodování dat dost přesně na to, abych byl schopen spolehlivě odhalit případné chyby.

7. Vhodné typy antén

K přijímači i vysílači je nezbytně nutné použít vhodnou anténu. Antén pro pásmo 433MHz je celá řada typů. Výrobce však doporučuje [1] následující typy.

7.1. Smyčková anténa

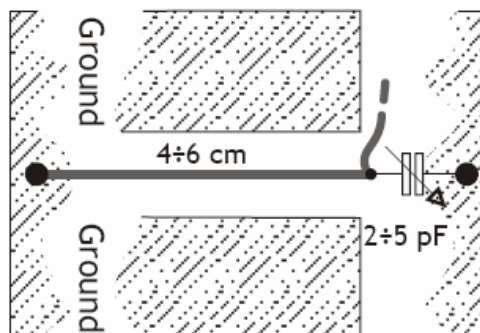
Smyčková anténa je velmi kompaktní, má rozměry cca 2 x 2 cm, nemusí být složitě nastavována. Jedná se o otevřenou smyčku o průměru 2 cm, která je na jednom konci přímo připojena na výstup vysílače a druhý konec je uzemněn. V porovnání s prutovou anténou je její účinnost výrazně nižší a to přibližně o 20 dB. Toho lze využít v aplikacích, kde nám jde především o rozměry a stačí nám nižší dosah vysílače. Některé moduly přijímačů a vysílačů mají tento typ antény přímo integrován.



Obr. 21: Smyčková anténa [1]

7.2. Rezonanční anténa

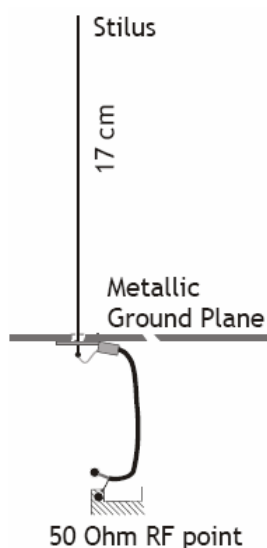
Tento typ antény lze také realizovat přímo na desce plošného spoje avšak vyžaduje proměnný kapacitor pro vyladění antény do rezonance. To znamená dosažení reálné impedance 50 Ω , což je i impedance výstupu vysílače (vstupu přijímače) a také impedance anténního vedení. Zářič je vytvořen ze 4 – 6 cm dlouhého vodiče. V porovnání s prutovou anténou je účinnost nižší o cca 10 dB.



Obr. 22: Rezonanční anténa [1]

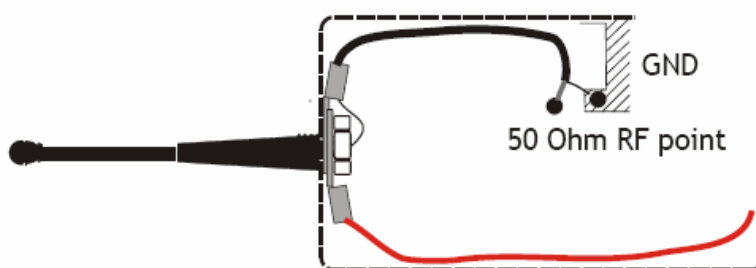
7.3. Vertikální prutová anténa umístěná na DPS

Anténa se vyznačuje jednoduchou mechanickou konstrukcí a zároveň nejvyšší účinností. Zářič musí být kolmý k zemní ploše desky plošných spojů a vyžaduje okolo sebe několik centimetrů volného prostoru. Anténu výrobce doporučuje pro většinu aplikací. Vzhledem k tomu, že má zisk 0 dB, je vhodné ji použít pro porovnávání jednotlivých typů modulů.



Obr. 23: Vertikální prutová anténa [1]

Pokud je třeba umístit anténu na nekovový povrch, například na panel plastové přístrojové skříňky, je nutné ji doplnit vhodnou protiváhou uvnitř skříňky. Pokud je zářič kratší, než 17 cm, klesá účinnost antény.



Obr. 24: Anténa s protiváhou [1]

Vzhledem k tomu, že se bude jednat o přenosné zařízení rozhodl jsem se použít prutovou anténu s protiváhou uvnitř krabičky přijímače i vysílače, kterou lze zakoupit hotovou.

8. Závěr

8.1. Shrnutí dosažených výsledků

Podařilo se navrhnout, zkonstruovat a oživit modul vysílače i přijímače bezdrátového příposlechu. Co se týče modulu vysílače, není zde žádný větší problém, který by bránil jeho praktickému používání. Snad jen o něco vyšší spotřeba v klidovém režimu, než bylo původně zamýšleno. V klidovém režimu odebírá modul vysílače (mikrokontrolér v módu Power down, hybridní modul vysílače odpojen) 3,6 mA. Během samotného vysílání stoupne odběr na 11,5 mA. Je to zapříčiněno tím, že jsem od prvotního návrhu nevěnoval příliš pozornosti obvodům napájení a použil nejjednodušší stabilizátor se Zenerovou diodou, přesto však je výdrž vysílače na baterie o kapacitě 800 mAh přibližně 150 h.

Ohledně přijímací části zařízení je situace poněkud horší. Nastal sice stejný problém s napájením, jako u vysílače, ale jeho řešení není tak očividné, jako v předchozím případě. Tento problém je navíc umocněn nemožností použití módu Power down, protože hybridní modul produkuje velké množství vlastního rušení, o kterém se výrobce v dokumentaci k modulu jednoduše vůbec nezmiňuje. Rušení je přenášeno z výstupu modulu přímo na vstup externího přerušení. Pomocí vhodně napsaného software se sice podařilo zajistit funkčnost zařízení, ale možnost jeho praktického použití je malá, protože odběr proudu v klidovém stavu dosahuje 8 mA. Pokud je zahájena signalizace poplachu a je aktivován zesilovač, bude spotřeba proudu ještě výrazně vyšší.

Během testování obou zařízení byly použity prutové antény o délce 17 cm přímo na desce plošných spojů. Dosah zařízení, kdy jsou ještě korektně přijata data, jsem naměřil přibližně 18 m v zástavbě (v cestě stály 3 suché, cihlové zdi standardní tloušťky) a kolem 100 m ve volném prostoru (pouze odhad vzdálenosti, nedisponuji pásmem). Toto jsou uspokojivé výsledky, vysílač navíc nepracuje na maximální možný výkon. Dosah je dostačující a zároveň není příliš velký, aby rušil podobná zařízení ostatních rybářů. Mechanická konstrukce obou částí ještě nebyla hlouběji řešena, protože podoba zařízení stále není definitivní.

V přijímací části jsem objevil ještě jeden nedostatek a tím jsou možnosti akustické a optické signalizace záběru. Jedná se o to, že provádět samotnou signalizaci lze jen pokud není aktivováno externí přerušení INT0, tudíž pouze pokud procesor nevyhodnocuje rušení nebo příchozí paket dat. Výsledkem je přerušovaná signalizace, což nepůsobí příliš uhlazeně.

8.2. Možnosti řešení vzniklých problémů

Jak jsem již uvedl v předchozích kapitolách, nejzávažnějším problémem je rušení, které produkuje hybridní modul přijímače Aurel BC-NBK. Vzhledem k tomu, že výrobce ve svých aplikačních poznámkách k modulu uvádí, že v době neaktivity je na výstupu modulu nízká úroveň, příliš jsem se tím nezabýval. Byla to chyba, kterou jsem

odhalil příliš pozdě. Z tohoto jasně vyplývá, že tyto levné moduly (z nabízeného sortimentu nejlevnější vůbec) nejsou vhodné pro aplikace napájené z baterií, kde je úspora energie na prvním místě. Jedná se především o přijímač, protože vysílač funguje přesně tak, jak je deklarováno.

Jako jednu z možností řešení tohoto problému jsem zkoušel externě nastavit komparátoru nižší prahovou úroveň, aby nebyl překlápěn šumem, ale to vedlo k drastickému snížení citlivosti přijímače a dosah se snížil přibližně o řád. Nastavení bylo navíc velice citlivé a nebylo snadné jej udržet delší dobu (řešením by bylo umístění součástek na desku plošných spojů).

Další možností by bylo vyvést signál z detektoru modulu do externího obvodu tvarovače. Tuto možnost jsem však netestoval, znamenala by totiž návrh a výrobu nové desky plošných spojů přijímače a na to jsem již neměl dostatek času.

Vzhledem k tomu, že toto zařízení hodlám používat, rozhodl jsem se pro jiné a podle mého názoru jistější řešení. Bude jím použití jiné, vhodnější dvojice hybridních modulů buď od stejného výrobce, nebo z produkce jiných firem.

Dalším problémem, který bude muset být vyřešen je napájení obou částí zařízení. Řešení spočívá v novém návrhu stabilizátorů napětí, jak je naznačeno v kapitolách 3.7. a 5.5.

Řešení posledního uvedeného problému spočívá v použití dalšího mikrokontroléru, který bude zajišťovat pouze provádění signalizace a bude ovládán stejným způsobem, jako nf zesilovač, tedy zapínán a vypínán hlavním mikrokontrolérem.

9. Použitá literatura

- [1] Aurel wireless, Oficiální stránky firmy Aurel [online]. 2004 – [cit. 20. května 2007]. Dostupné na WWW: <http://www.aurel.it>
- [2] Atmel, Oficiální stránky firmy Atmel [online]. 2008 – [cit. 20. května 2008]. Dostupné na WWW: <http://www.atmel.com>
- [3] ATtiny84V 8bit AVR mikroprocesor, Data sheet [online]. Atmel Corp., 2006 – [cit. 20. května 2008]. Dostupné na WWW: <http://www.atmel.com>
- [4] TX – SAW 433/s – z transmitter, Instruction manual [online]. Aurel. 2002 – [cit. 20. května 2008]. Dostupné na WWW: <http://www.aurel.it>
- [5] BC – NBK receiver, Instruction manual [online]. Aurel. 2002 – [cit. 20. května 2008]. Dostupné na WWW: <http://www.aurel.it>
- [6] CadSoft Computer GmbH, Oficiální stránky firmy Cadsoft [online]. 2007 – [cit. 20. května 2008]. Dostupné na WWW: <http://www.cadsoft.de>
- [7] Hrbáček, Jiří. Komunikace mikrokontroléru s okolím 2. BEN, Praha, 2000. ISBN 8086056732
- [8] LM386 Audio power amplifier, Data sheet [online]. National Semiconductor. 1993 – [cit. 20. května 2008]. Dostupné na WWW: <http://www.national.com>
- [9] Dufek, Ivo. Příposlech k rybářskému hlásiči v pásmu LPD: Semestrální projekt 2 (BB2E). FEKT VUT v Brně, 2008, 26 s.
- [10] Mikroelektronika, Oficiální stránky firmy Mikroelektronika [online]. 2008 – [cit. 20. května 2008]. Dostupné na WWW: <http://www.mikroe.com>
- [11] HW server, Oficiální stránky firmy HW server s.r.o. [online]. 2008 – [cit. 20. května 2008]. Dostupné na WWW: <http://www.hw.cz>

Přílohy

A Seznam součástek vysílače

C1, C2, C3	330nF	C ker. SMD 1206
C4, C5	10nF	C ker. SMD 1206
C6, C7	22pF	C ker. SMD 1206
C8	10 μ F / 16V	C elektrolyt
C9, C10	100nF	C ker. SMD 1206
C8	10 μ F / 16V	C elektrolyt
D1	BZX55 4V2	SOD80C
D2	LED červená	LED 3mm
IC1	ATtiny84V	IC DIL14
IC2	4012D	IC SOIC14
Q1	2MHz	krystal HC49 U
Q2	BS250	P – MOSFET TO – 92
Q3, Q4, Q5	BC848A	SOT23
R1, R2, R3	6k8	R SMD 1206
R4, R5, R6	680	R SMD 1206
R7	10k	R SMD 1206
R8	1k2	R SMD 1206
R9	4k7	R SMD 1206
ISP	MA03 – 2	konektor ISP
TX1	TX – SAW433	modul TX Aurel

B Seznam součástek přijímače

C1, C2	27pF	C ker. SMD 0805
C3, C5	10nF	C ker. SMD 0805
C4	47nF	C ker. SMD 0805
C6, C7	100nF	C ker. SMD 0805
C8	100 μ F / 16V	C elektrolyt
C9, C10	10 μ F / 16V	C elektrolyt
C11	220 μ F / 16V	C elektrolyt
C12	1 μ F / 16V	C elektrolyt
D1	BZX55 4V2	SOD80C
D2	LED červená	LED 3mm
D3	LED zelená	LED 3mm
D4	LED žlutá	LED 3mm
IC1	ATtiny84V	IC DIL14
IC2	LM386	IC DIL08
Q1	2MHz	krystal 2MHz HC49 U
Q2	BS250	P – MOSFET TO – 92
R1	1k2	R SMD 1206
R2	1 Ω	R SMD 1206
R3	10k	R SMD 1206
R4	1k2	R SMD 1206
R5, R6, R7	270	R SMD 1206
R8	4k7	R SMD 1206
RX1	RX – BC – BNK	modul RX Aurel
SV1	MA03 – 2	konektor ISP
SV2	MA03 – 1	kon. potenciometr hlasitosti

Seznam zkratek

ASK	Amplitude Shift Keying
CRC	Cyclic Redundancy Check
DPS	deska s plošnými spoji
FEC	Forward Error Correction
FSK	Frequency Shift Keying
ISP	In-System Programming
OOK	On-Off Keying
RX	příjem, přijímací
SMD	Surface Mounted Device
TX	vysílání, vysílací
VF	vysokofrekvenční
NF	nízkofrekvenční