



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

JEDNOSMĚRNÁ SÉRIOVÁ KOMUNIKACE LASEREM NA VĚTŠÍ VZDÁLENOST

ONE-WAY SERIAL LASER COMMUNICATION OVER LONGER DISTANCES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Adam Valent

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Jiří Háze, Ph.D.

BRNO 2021



Diplomová práce

magisterský navazující studijní program **Mikroelektronika**

Ústav mikroelektroniky

Student: Bc. Adam Valent

ID: 195458

Ročník: 2

Akademický rok: 2020/21

NÁZEV TÉMATU:

Jednosměrná sériová komunikace laserem na větší vzdálenost

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte problematiku sériového přenosu a amplitudové modulace signálu. Na základě získaných poznatků sestrojte zařízení na jednosměrný přenos dat mezi dvěma zařízeními na více než 10 metrů. Jako vysílač použijte LASER diodu a jako přijímač fotovoltaický panel. Vysílač a přijímač budou připojitelné k počítači. Vytvořte obslužný program pro odesílání a přijímání dat. Zařízení realizujte s minimální chybovostí přenosu a minimální rychlostí 1 kB/s.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce.

Termín zadání: 8.2.2021

Termín odevzdání: 25.5.2021

Vedoucí práce: doc. Ing. Jiří Háze, Ph.D.

doc. Ing. Lukáš Fojcik, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

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

ABSTRAKT

Predmetom tejto diplomovej práce je konštrukcia zariadenia na jednosmerný prenos údajov. Toto zariadenie sa skladá z vysielачa a prijímača, z ktorých je každý pripojený k vlastnému počítaču USB rozhraním. Pomocou neho je možné posielat' UTF-8 znaky, alebo súbory z jedného počítača na druhý. Na oboch počítačoch je spustený obslužný program. Jadrom vysielачa je laserová dióda modulujúca digitálny signál. Prijímačom je solárny panel s rezonančným obvodom a zosilňovačom. Komunikáciu medzi elektronikou a počítačom zabezpečujú mikrokontroléry. Prijaté dáta sú overované jedným z viacerých overovacích algoritmov, zvoliteľných užívateľom v obslužnom programe.

KLÚČOVÉ SLOVÁ

prenos údajov, sériová komunikácia, laserová dióda, fotovoltaický panel, algoritmy na detekciu chýb, mikrokontrolér

ABSTRACT

The subject of this thesis is the construction of one-way communication device. This device consists of the transmitter and the receiver, both of which are connected to its respective computer via USB interface. This device allows sending UTF-8 characters or files from one computer to another. Both computers are running a graphical user interface program. The core of a transmitter is a digital signal modulating laser diode. The receiver is made of photovoltaic panel with a resonance circuit and an amplifier. Communication between the electronics and the computer is driven by microcontrollers. Received messages are verified with one of multiple error detection algorithms, which can be selected by user in the utility program.

KEYWORDS

data transmission, serial communication, laser diode, photovoltaic diode, error detection algorithms, microcontroller

VALENT, Adam. Jednosměrná sériová komunikace laserem na větší vzdálenost [online]. Brno, 2021 [cit. 2021-05-03]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/134655>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav mikroelektroniky. Vedoucí práce Jiří Háze.

PREHLÁSENIE

Meno a priezvisko študenta: *Adam Valent*
VUT ID študenta: *195458*
Typ práce: *Diplomová práca*
Akademický rok: *2020/2021*
Zadanie záverečné práce: *Jednosmerná sériová komunikácia laserom na väčšiu vzdialenosť*

Prehlasujem, že svoju záverečnú prácu som vypracoval samostatne, pod vedením docenta Jiřího Háze a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej záverečnej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121 / 2000 Sb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 českého Trestného zákonníka č. 40/2009 Zb.

V Brne dňa: 25. mája 2021

podpis autora

POĎAKOVANIE

Ďakujem vedúcemu diplomovej práce, docentovi Ing. Jiřímu Háze Ph.D., za účinnú metodickú, pedagogickú a odbornú pomoc a ďalšie cenné rady pri spracovaní mojej diplomovej práce. Taktiež som vďačný za ochotu, rady a korekciu formálnej stránky tejto diplomovej práce.

V Brne dňa: 25. mája 2021

podpis autora

OBSAH

ÚVOD	1
1 TEÓRIA HARDVÉROVEJ ČASTI.....	3
1.1 Fotovoltaický efekt.....	3
1.1.1 Rýchle a pomalé nosiče	5
1.1.2 Kapacita diódy	6
1.2 Lasery	6
1.2.1 Čas zapnutia	7
1.3 Rezonančné obvody	8
1.4 Tranzistorový zosilňovač	9
1.4.1 Zapojenie so spoločným emitorom.....	9
2 TEÓRIA SPOĽAHLIVÉHO PRENOSU	12
2.1 Parita	14
2.2 Kontrolný súčet	14
2.3 Hammingov kontrolný súčet	16
2.3.1 Implementácia do softvéru.....	18
2.4 Cyklický redundantný súčet.....	19
2.4.1 Implementácia do hardvéru	25
2.4.2 CRC modifikácie	26
3 NÁVRH ZARIADENIA	27
3.1 Schéma paketu	28
3.2 Synchronizácia	30
3.3 Komunikácia s počítačom	31
3.4 Uživatelské rozhranie	32
3.5 Prenos bajtov	33
3.6 Prenos súborov	33
3.6.1 Štruktúra hlavičky.....	35

3.7	VRSTVY ZARIADENIA	36
4	REALIZÁCIA HARDVÉROVEJ ČASTI	38
4.1	Laserová dióda	38
4.2	Elektronika vysielача	38
4.3	Solárny panel.....	41
4.4	Elektronika prijímača	44
4.5	Krabičky	56
4.5.1	Krabička prijímača.....	56
4.5.2	Krabička vysielача.....	57
5	REALIZÁCIA SOFTVÉROVEJ ČASTI.....	59
5.1	Signálová vrstva	59
5.1.1	Prijímač.....	60
5.1.2	Vysielач	62
5.2	Dátová vrstva	64
5.2.1	Parita	64
5.2.2	Kontrolný súčet.....	64
5.2.3	Hammingov kód	64
5.2.4	Cyklický redundantný súčet.....	65
5.3	Transportná vrstva.....	66
5.4	Aplikačná vrstva.....	68
5.5	Grafické rozhranie.....	70
6	TESTOVANIE	74
6.1	Rýchlosť prenosu	74
6.2	Dosah prenosu.....	74
6.3	Svetelná saturácia.....	74
6.4	Odolnosť voči chybám.....	75
6.5	Cena.....	75

ZÁVER	76
LITERATÚRA	78
ZOZNAM OBRÁZKOV	82
ZOZNAM TABULIEK	84
PRÍLOHY	85

ÚVOD

Prenos údajov je neodmysliteľná časť modernej civilizácie. Či už to je internet, komunikácia medzi družicami alebo aj komunikácia medzi čipmi na doske plošného spoja. Bez rýchlej výmeny údajov by svet aký je neexistoval. Trendom je posúvanie hraníc maximálnej rýchlosti dopredu pre ešte lepší zážitok pri pozeraní živých videí, hraní hier, alebo pre plynulé prehliadanie webových stránok. Okrem toho je tu spoľahlivosť prenosu a algoritmy, ktoré overujú prijaté údaje. Tieto chyby odhaľujúce algoritmy nesmú príliš zdržovať výmenu údajov, ale zároveň musia s vysokou pravdepodobnosťou odhaľovať poškodené dáta. Preto je potrebné spraviť prenos maximálne optimalizovaný pre túto úlohu a odolný voči náhodným chybám a šumu.

Cieľom tejto diplomovej práce je vyrobiť zariadenie pozostávajúce z prijímača a vysielača na jednosmerný prenos údajov medzi dvoma počítačmi. Vysielač predáva údaje pomocou laserového lúča dopadajúceho na prijímač. Zdrojom laserového lúča je laserová dióda, ktorá vysiela modulovaný signál. Prijímačom je fotovoltaiický panel. Moduláciu a demoduláciu má na starosti samostatná elektronika. Prijímač aj vysielač obsahuje mikrokontrolér, ktorý vytvorí virtuálny sériový port na prenos údajov medzi počítačom. Zariadenie je napájané z USB. Na počítači pripojeného k prijímaču je program, ktorý zobrazuje prijaté údaje. Na počítači pripojeného k vysielaču je program, do ktorého sa zadávajú údaje k prenosu. Z obslužného programu je možné posielat dáta v textovom (UTF-8), alebo v binárnom (súbory) móde. Prijaté údaje kontroluje overovací algoritmus, ktorý je možné zvoliť. Algoritmami na výber sú parita, kontrolný súčet, Hammingov kód a cyklický redundantný súčet (CRC). Najzložitejší je CRC algoritmus, ten sa používa v technológiách, ako je napríklad Bluetooth, alebo Ethernet. Vybratý algoritmus musí byť rovnaký na oboch počítačoch. Zámerom je dosiahnuť čo najväčšiu rýchlosť (minimálne 1 kB/s) a minimálnu chybovosť. Prenos by mal bez problémov fungovať aspoň na 10 metrov. Pre každý algoritmus bude vyhodnotená spoľahlivosť a efektívnosť. Na konci práce bude doporučené, ktorý z algoritmov sa najviac hodí do tejto aplikácie a praktické použitie takéhoto zariadenia.

Táto práca zasahuje do viacerých odborných oblastí, od návrhu, až po praktické vyhotovenie. V prvej teoretickej časti sú prebraté základy overovania chyby odhaľujúcimi algoritmami pre lepšie pochopenie, čo bude potrebné implementovať do programu mikrokontroléru. V druhej, teoretickej časti, sú prebraté fyzikálne princípy

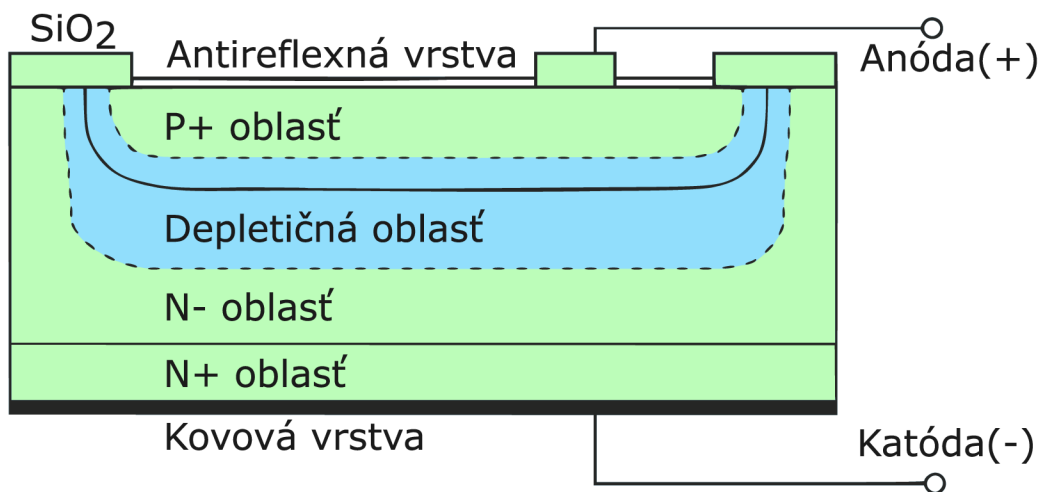
fotovoltaického panelu a laserovej diódy. Fotovoltaický panel nie je primárne určený na prenos údajov, preto je žiadúce čo najviac pochopiť jeho vnútornú štruktúru a vlastnosti, aby sa z neho vytlačilo maximum a bolo možné splniť minimálnu požiadavku na rýchlosť prenosu. Je lacný a má veľkú plochu, čo je jeho výhoda. Za týmto nasledujú praktické časti. Ako prvá je analyzovaná laserová dióda a navrhnutá elektronika modulátora. Základ modulátora je kruhový oscilátor a prúdový zdroj. Pokračuje sa analýzou solárneho panelu a návrhom elektroniky demodulátora. Základ tohto obvodu je rezonančný obvod so zesilňovačom a komparátorom. S vyrobeným modulátorom a demodulátorom je možné navrhnuť krabičky pre obe zariadenia. Krabičky sú vytlačené na 3D tlačiarňi. Predposlednou časťou je implementácia chyby overovacích algoritmov do mikrokontroléru a tvorba obslužného programu. Poslednou časťou je testovanie výrobku. Zariadenie bude podrobené viacerým testom. Kľúčové pri týchto testoch je odhaliť limity zariadenia, poprípade jeho nedostatky.

1 TEÓRIA HARDVÉROVEJ ČASTI

Táto kapitola je venovaná teórii elektroniky, ktorú je potrebné ovládať k zostrojeniu laserového vysieláča a fotovoltaiického prijímača. V prvom rade sa bude zaoberať teoretickým základom premeny elektrickej energie na svetelnú a naopak, a následne súvisiacim fyzikálnym javom a princípmi v polovodičoch. Neskôr sú v kapitole spomenuté aj rezonančné obvody, ktoré určujú frekvenciu a šírku pásma, v ktorej sa bude komunikovať.

1.1 Fotovoltaiický efekt

Premena svetelnej energie na elektrickú sa deje za pomoci fotoelektrického javu, ktorý prebieha vo fotovoltaiickom článku. Fotovoltaiický článok je spojenie kremíku s rôznym typom vodivosti. Ide v podstate o veľkoplošnú diódu, ktorej vlastnosti sú upravené tak, aby mala čo najvyššiu efektivitu premeny svetelnej energie na elektrickú. Na obrázku 1.1 je vidieť vnútornú štruktúru článku [1]. Typicky je na vrchu oblasť vodivosti typu P, ktorá je intenzívne dopovaná. Pod ňou je oblasť vyprázdneného náboja (depletičná oblasť), kde sa vytvorí elektrické pole a za ňou je oblasť typu N, málo dotovaná. Rozdiel v dotácii spôsobí to, že sa elektrické pole rozťahne hlbšie do N oblasti. Toto je žiadúce, keďže elektrické pole následne rozdeľuje páry elektrón-diera vytvorené fotoelektrickým javom. Je to dobre, keď oblasť v ktorej pôsobí, je čo najväčšia. Keďže oblasť P je silne dopovaná, stačí aj jej tenká vrstva na to, aby vytiahla elektróny z N oblasti a zaručila tak širokú vyprázdnenú oblasť. Oblasť typu P nesmie byť hrubá, aby cez ňu prenikalo čo najviac svetla do prechodu.



Obrázok 1.1 Prierez štruktúrou solárneho panelu [1]

Pri bežných teplotách zemského povrchu sú všetky prímеси v kremíku ionizované, preto na vytvorenie páru elektrón-diera je potrebná energia pre ionizáciu čistého kremíku, ktorá je 1,1 eV. Je teda možné vypočítať minimálnu vlnovú dĺžku λ , od ktorej bude prebiehať premena svetelnej energie na elektrickú [2]

$$E[J] = E[eV] \cdot q = 1,1 \cdot 1,602 \cdot 10^{-19} = 1.762 \cdot 10^{-19} J,$$

$$\lambda = \frac{h \cdot c}{E[J]} = \frac{6.626 \cdot 10^{-34} \cdot 3 \cdot 10^8}{1.762 \cdot 10^{-19}} \approx 1128 \text{ nm}, \quad (1.1)$$

kde q je náboj elektrónu, h je Planckova konštanta a c je rýchlosť svetla. Vlnová dĺžka energie 1,1 eV zodpovedá približne 1128 nm, ktorá je v spektre blízkej infračervenej oblasti. Pre dlhšie vlnové dĺžky, respektíve menšie energie, sa kremík stáva priehľadným. Pre zisk elektrickej energie zo svetla sa avšak nedá ísť ani do vysokých energií. Takýto druh žiarenia sa pohltí tesne pod povrchom a nedostane sa do vyprázdnenej oblasti. Nehovoriac o tom, že väčšina článkov má hornú vrstvu z polyméru, ktorá chráni kremíkový kryštál. Takmer vždy je táto vrstva nepriepustná pre vysoké energie žiarenia, ak nejde o špeciálu fotodiódu. Hranica maximálnej energie je okolo 4.13 eV, čo zodpovedá 300 nm. Obrázok 1.2 popisuje spektrálnu odozvu článku [3]. Prúd diódou je braný na jednotku jej plochy (Am^{-2}), tak isto aj optický výkon osvetľujúci diódu (Wm^{-2}). Z toho výsledná jednotka AW^{-1} . V tomto prípade je solárny panel oveľa väčší, ako plocha, ktorú osvetľuje laser. Preto sa obe plochy rovnajú a senzitivita je jednoducho prúd panelu I podelený výkonom laseru P [8]

$$R_\lambda = \frac{I}{P} \quad [\text{AW}^{-1}]. \quad (1.2)$$

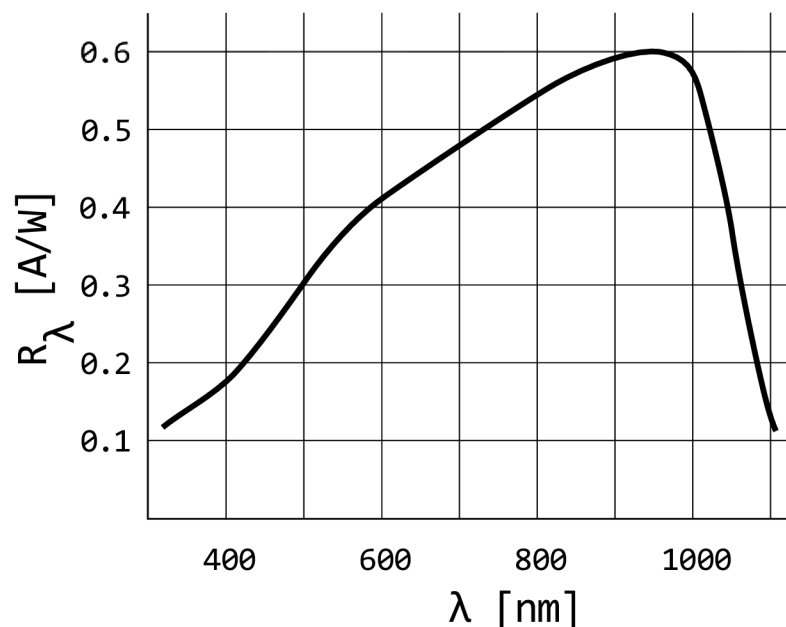
Za zmienku stojí aj vonkajšia kvantová efektívnosť EQE (z angličtiny „*external quantum efficiency*“), ktorá vyjadruje pomer medzi užitočnými fotónmi (tie ktoré odovzdali svoju energiu elektrónom a prispeli k vedeniu prúdu) a všetkými fotónmi dopadajúcimi na solárny panel za jednotku času [8]

$$EQE = R_\lambda \frac{h \cdot c}{\lambda \cdot q} \quad [-]. \quad (1.3)$$

Existuje aj vnútorná kvantová efektívnosť IQE . Tá je vždy väčšia ako vonkajšia (teoreticky sa môže rovnať EQE), ktorá hovorí koľko z absorbovaných fotónov prispelo k vedeniu prúdu. Teda nepočíta odrazené (neabsorbované) fotóny [19].

Toto ale nie je jediný problém efektivity. Pretože ak sa fotón s vyššou energiou ako 1,1 eV dostane do vyprázdnenej oblasti, vyrazí síce elektrón čím vytvorí pár elektrón-diera. Prebytočnú energiu nad 1.1 eV však neudrží a stratí ju takmer okamžite pri zrážkach s kryštálovou mriežkou [4]. Tieto nosiče sa nazývajú „horúce nosiče“, pretože týmito zrážkami ohrievajú kremík. Aby novovytvorené nosiče prispeli k driftovému prúdu, musí ich fotón rozdeliť v priestore vyprázdneného náboja. Tam ich od seba odtiahne elektrické pole. Preto je dôležité mať túto oblasť čo najširšiu.

Maximálne napätie jedného fotovoltaického panelu je približne 0,5 V. Jeho hodnotu určuje sila elektrického poľa vyprázdnenej oblasti. Preto je väčšina solárnych panelov zložená v sérii z viacerých fotovoltaických článkov.



Obrázok 1.2 Senzitivita kremíkovej fotodiódy [6]

1.1.1 Rýchle a pomalé nosiče

Ak fotóny vyrážajú elektróny z valenčného pásma v oblasti vyprázdneného náboja, elektrické pole okamžite rozdelí páry elektrón-diera, ktoré prispejú k celkovému prúdu. V takomto prípade ide o rýchle nosiče respektíve o driftový prúd. Avšak nie všetky generačné procesy sa odohrávajú v depletičnej oblasti. Niektoré fotóny vyrazia elektrón v neutrálnej oblasti N alebo P. V takomto prípade, ak sa im nepodarí zrekombinovať, musia predifundovať cez mriežku kremíku, kým sa nedostanú do vyprázdnenej oblasti, v ktorej pôsobí elektrické pole. Tieto nosiče sú veľmi pomalé a ide o difúzny prúd. Pre danie pomeru rýchlosti do perspektívy, čas na presun elektrónu driftom je 100 nm za jednu ps. Zatiaľ čo difúziou prejde iba 1,25 nm za jednu ps [16].

1.1.2 Kapacita diódy

Prechod PN má kapacitu danú dvoma dielčimi kapacitami. Prvá je kapacita depletičnej oblasti. Je možné si ju predstaviť ako kapacitu doskového kondenzátoru, kde vzdialenosť medzi doskami je šírka depletičnej oblasti W . Zatiaľ čo plocha dosiek A je daná prierezom diódy. Táto kapacita je dominantná v závernom smere a zväčšuje sa so zmenšujúcou sa šírkou depletičnej oblasti. Jej hodnota je daná vzťahom [5]

$$C_{DEP} = \varepsilon \frac{A}{W(U)} \quad [F], \quad (1.4)$$

kde ε je permitivita kremíku. Šírka oblasti je funkcia napätia, kde napätie proti smeru elektrického pola tlačí dve pomyselné dosky kondenzátora k sebe. Druhá kapacita je dominantná v priepustnom smere a je daná množstvom náboja Q , ktorý prechádza diódou na jednotku napätia U . Nazýva sa difúzna kapacita a počíta sa podľa vzťahu [5]

$$C_{DIF} = \frac{dQ}{dU} = \frac{dI \cdot \tau}{dU} \quad [F], \quad (1.5)$$

kde τ je stredná doba života nosičov v dióde. Táto kapacita je hlavne závislá od pretekajúceho prúdu, keďže napätie na dióde od prekročenia difúzneho napätia sa málo mení. Dióda zapojená v závernom smere má teda zanedbateľnú difúznu kapacitu. So zvyšovaním záverného napätia klesá aj depletičná kapacita [7].

1.2 Lasery

Lasery fungujú na princípe zosilňovania svetla v aktívnej látke. Lasery je možné rozdeliť podľa skupenstva aktívnej látky alebo druhu budenia. Pre vysielač je vhodný predovšetkým pevno-látkový polovodičový laser. Tieto lasery sú priamo budené elektrickým prúdom. Výhodou polovodičových laserov je, že sú malé, keďže pevná aktívna látka má vysokú hustotu atómov. K zosilneniu potom dochádza na krátkej vzdialenosti. Ide v podstate o špeciálne upravenú LED diódu. Ak sa laserová dióda pripojí na elektrický zdroj, ktorý predá dostatočnú energiu elektrónom a dieram, začnú voľné nosiče rekombinovať v strede PN prechodu. Väčšina nosičov zrekombinuje a predá energiu vo forme fonónu (vibrácií kryštálovej mriežky), avšak určité množstvo zrekombinuje vo svetelný prechod a vytvorí fotón. Frekvencia vyžiareného fotónu (teda jeho energia) sa rovná energetickému rozdielu medzi hladinami elektrónu a diery. Na

krajoch štruktúry sú zrkadlá, ktoré odrážajú časť fotónov naspäť dovnútra, kde znovu pomáhajú rekombinácií. Tomuto javu sa hovorí stimulovaná emisia [14]. Moderné lasery sa vyrábajú s dvojitou hetero-štruktúrou, čo znamená, že určité miesto v strede laseru je z iného polovodičového materiálu a šírka zakázaného pásu je v tomto mieste užšia. Nosiče majú tendenciu rekombinovať tam a nie niekde po okrajoch, čím efektívnejšie prispievajú k stimulovanej emisii. Zároveň je presne definovaný objem, v ktorom dochádza ku rekombinácií, z čoho je možné vypočítať optický výkon.

Aby laser fungoval, potrebuje dosiahnuť inverzné obsadenie hladín, keďže bez neho sa laser utlmí tým, že vyžiarené fotóny budú narážať do elektrónov na spodnej hladine. Pre toto laser nevyžaruje lúč, ak ním tečie nízky prúd a veľa elektrónov je stále na spodnej hladine. Prebieha ale takzvaná spontánna emisia, kedy elektróny padajú na spodnú hladinu a generujú fotóny s náhodným smerom (tak isto ako LED dióda). So zvyšujúcim sa prúdom bude čím ďalej tým menej elektrónov na spodnej (nulte) hladine. Na druhej strane, bude čoraz viac fotónov čakať na prvej hladine, z kadiaľ ich stiahne stimulovaná emisia a laser začne vyžarovať koherentný lúč.

Keďže je polovodičový laser skonštruovaný ako prechod PN, má rovnaké elektrické vlastnosti ako dióda. Je teda nutné použiť prúdovú limitáciu na jeho obsluhu. V najhoršom prípade rezistor, v najlepšom prípade prúdový zdroj, pretože so zvyšujúcou sa teplotou má jeho dynamický odpor tendenciu klesať. Aby sa laserový lúč nerozchádzal, potrebuje šošovku. To platí pre každý polovodičový laser, pretože bez šošovky sa svetlo z neho môže rozchádzať až 30° od osy. Väčšinu optík polovodičových laserov je možné manuálne dostaviť.

1.2.1 Čas zapnutia

Čas zapnutia (z angličtiny „*time on delay*“, skratka T_{on}). Je to čas, ktorý uplynie od začiatku privedenia napätia, po čas kedy začne laser vyžarovať. Je daný súčtom dvoch časov. Vyžarovanie začína pri vytvorení inverzného obsadenia hladín, kedy nedochádza k častým zrážkam s elektrónmi na spodnej hladine. Elektróny majú dobu života, s ktorou sa držia vo vodivostnom pásme, kým ich nestiahne dole fotón, alebo nezrekombinujú spontánne. K tomuto času sa ešte pripočíta čas nabíjania parazitného kondenzátoru laseru. Pretože ide o diódu, kapacitu má tú istú ako dióda fotovoltického článku (1.4) a (1.5). Čas doby života elektrónu vo vodivostnom pásme vyrobenej diódy nie je možné ovplyvniť, ale je možné zmeniť veľkosť prúdu, ktorý nabíja kondenzátor. Preto sa

niekedy pri zapínaní púšťa do laseru väčší prúd pre zníženie tohto času. Ak ide o moduláciu a rýchle spínanie laseru, nie je dobré ísť pod hraničnú hodnotu prúdu, pri ktorej laser prestáva vyžarovať, aby sa nemusel parazitný kondenzátor zbytočne vybíjať a nabíjať a aby laser nestratil inverzné obsadenie hladín.

1.3 Rezonančné obvody

Rezonančný obvod je spojenie kapacity C , indukčnosti L a odporu R , ktoré vykazujú iba reálnu zložku impedancie pri rezonančnej frekvencii. Kapacitná a indukčná reaktancia sa počíta podľa vzťahu [12] (v niektorej literatúre sa uvádza ako kapacitancia a induktancia)

$$X_c = \frac{1}{\omega C j} \quad [\Omega] \quad X_L = \omega L j \quad [\Omega], \quad (1.6)$$

kde ω je uhlová rýchlosť $2\pi f$ a j je imaginárna jednotka. Rezonančná frekvencia sa počíta podľa Thomsonovho vzťahu [12]

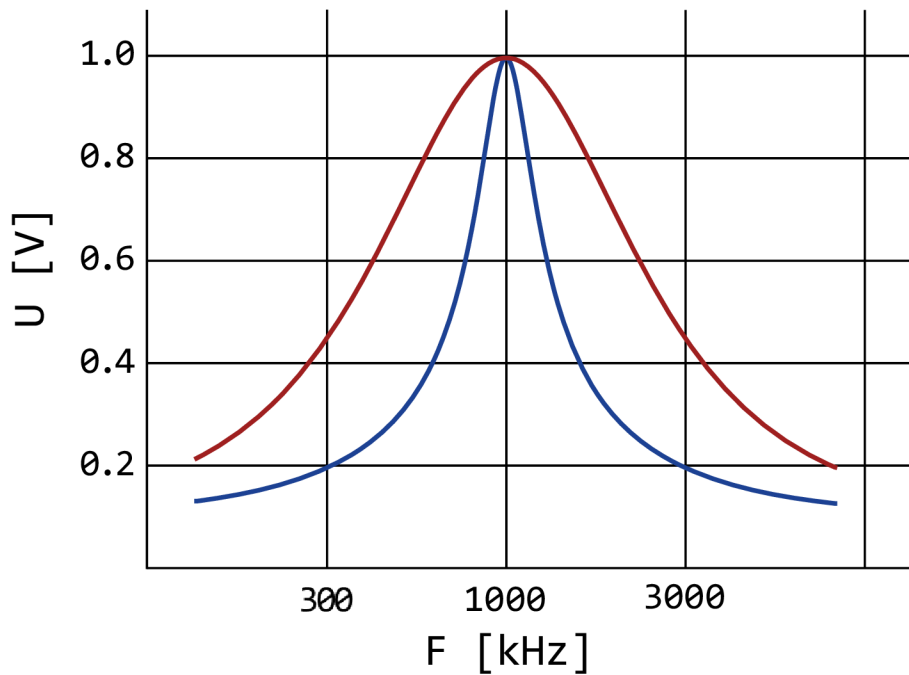
$$f = \frac{1}{2\pi\sqrt{LC}} \quad [Hz]. \quad (1.7)$$

V rezonancií sa hodnota kapacitnej reaktancie vyruší s indukčnou reaktanciou a zapojenie má iba odporový charakter. Pri týchto obvodoch je možné zistiť faktor kvality. Ten hovorí o selektivite obvodu v okolí rezonancie. Obvody s nízkym faktorom kvality majú krátku zotrvačnosť a širokú selektívnosť. Naopak, obvody s vysokým faktorom kvality majú dlhú zotrvačnosť a úzku selektívnosť. Podľa spojenia indukčnosti a kapacity sa rezonančné obvody rozdeľujú na sériové a paralelné. Pre sériový rezonančný obvod sa faktor kvality počíta podľa vzorca

$$Q_{ser} = \frac{1}{R} \sqrt{\frac{L}{C}} \quad [-]. \quad (1.8)$$

Faktor kvality je bezrozmerné číslo. Pre paralelný rezonančný obvod sa počíta ako

$$Q_{par} = R \sqrt{\frac{C}{L}} \quad [-]. \quad (1.9)$$



Obrázok 1.3 Dva rôzne faktory kvality paralelného rezonančného obvodu

Dva rôzne faktory kvality paralelného rezonančného obvodu sú na obrázku 2.1. Modrý rezonančný obvod má vyšší faktor kvality, ako červený rezonančný obvod.

1.4 Tranzistorový zosilňovač

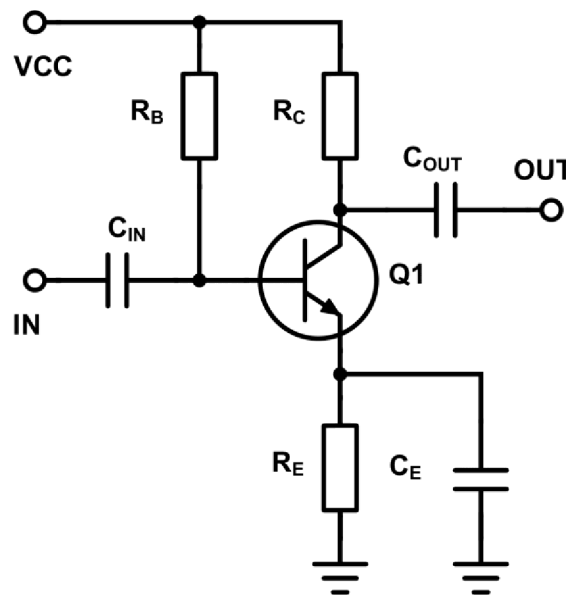
Ak ide o bipolárny tranzistor, existujú 3 spôsoby ako ho zapojiť do obvodu. Je to zapojenie so spoločným emitorom, so spoločnou bázou a so spoločným kolektorom. Každé zapojenie má svoje unikátne vlastnosti. Ich výhodou oproti operačným zosilňovačom je ich nižšia cena a vyššia rýchlosť. Nevýhodou je ich nelinearita. Keď sa vstupný signál pohybuje v malom rozsahu, sú ideálnym riešením zosilnenia.

1.4.1 Zapojenie so spoločným emitorom

Najčastejšie používané je zapojenie so spoločným emitorom. Má menšie skreslenie a najväčšie výkonové zosilnenie. Striedavý signál sa privádza na bázu cez oddeľovací kondenzátor, kde sa pripočíta, respektíve odpočíta, od prúdu tečúceho do báze. To sa následne odzrkadlí na prúde pretekajúcom kolektorom, čo vyvolá zmenu napätia na kolektorovom odpore. Napät'ové zosilnenie sa počíta ako [9]

$$A_U = \frac{U_{OUT}}{U_{IN}} = \frac{U_C}{U_B} = \frac{I_E R_C}{I_E r'_E} = \frac{R_C}{r'_E} \quad [-]. \quad (1.10)$$

Často krát sa do emitoru pridáva RC člen. Ak sa β zmení kvôli teplote, napätie na emitore porastie a jemne sa zmenší U_{BE} , čím sa prúd I_C nezvýši tak rapídne. Čím je odpor R v RC člene väčší, tým je tento efekt silnejší, avšak záleží aj na tranzistore. Ak by dióda medzi bázou a emitorom bola dokonalá, napätie na nej by sa nemenilo a žiadna stabilizácia by neprebehla. Na druhú stranu zvyšovaním odporu R_E klesá prúd tranzistorom, čím sa znižuje maximálna pracovná frekvencia. V praxi treba nájsť vždy dobrý pomer medzi stabilizáciou a zosilnením.



Obrázok 1.4 Zapojenie BJT so spoločným emitorom

Za zmienku stojí aj vstupná a výstupná impedancia. Ideálne by mala mať vstupná impedancia nekonečný odpor a výstupná impedancia nulový odpor. V skutočnosti to ale tak nie je. Často skutočná vstupná alebo výstupná impedancia znižuje zosilnenie a niekedy aj výrazne. Vstupná impedancia sa počíta podľa vzťahu

$$Z_{IN} = \frac{1}{\frac{1}{\frac{1}{2\pi f \cdot C_M} + R_C} + \frac{1}{R_B} + \frac{1}{\beta \cdot r'_e}} \quad [\Omega]. \quad (1.11)$$

Kde C_M je Millerova kapacita. Millerova kapacita je obyčajná parazitná kapacita medzi bázou a kolektorom, vynásobená napäťovým zosilnením. Keďže tranzistor so spoločným emitorom otáča fázu, napätia medzi bázou a kolektorom idú oproti sebe. Preto sa kondenzátor medzi týmito uzlami javí s násobne väčšou hodnotou. Tomuto javu je možné predísť takzvaným kaskodovým zapojením. Kedy sa medzi kolektor a kolektorový odpor

vsunie ďalší tranzistor. Ten pracuje v režime spoločnej báze a teda zosilňuje napätie na jeho emitore. Pôvodný tranzistor je potom nastavený na malé zosilnenie, aby Millerova kapacita bola čo najmenšia. Problém veľkej kapacity medzi bázou a kolektorom je znemožnenie zosilnenia vo vysokých frekvenciách, pretože vnútorný odpor zdroja a Millerova kapacita tvoria RC článok.

Okrem problému s Millerovou kapacitou, je tu problém so šumom. V bipolárnom tranzistore sú 3 zdroje šumu. Šum odporu r_b báze

$$U_{nrb} = \sqrt{4kT \cdot r_b} \quad [V/\sqrt{Hz}], \quad (1.12)$$

prúdový šum kolektora

$$I_{nc} = \sqrt{2q \cdot I_c} \quad [A/\sqrt{Hz}], \quad (1.13)$$

a posledný, prúdový šum báze

$$I_{nb} = \sqrt{2q \cdot I_b} \quad [A/\sqrt{Hz}], \quad (1.14)$$

ktorý sa často zanedbáva, ak tranzistor napája zdroj s malým vnútorným odporom. Pretože šumový prúd na ňom vytvorí zanedbateľne malé napätie oproti šumu odporu báze. Šumové príspevky sa prepočítavajú do jedného uzlu, kde sa sčítajú nekorelované, pretože šum je náhodný jav. Nekorelovaný súčet je v podstate diagonála N dimenzionálneho telesa, kde veľkosti strany telesa sú jednotlivé príspevky. Napäťový šum z N príspevkov by sa vypočítal ako

$$U_{ntot} = \sqrt{U_{n1}^2 + U_{n2}^2 \dots U_{nN}^2} \quad [V/\sqrt{Hz}]. \quad (1.15)$$

2 TEÓRIA SPOĽAHLIVÉHO PRENOSU

Táto kapitola je venovaná teórii prenosu, prijímaniu dát, spracovaniu dát a odhaľovaniu chýb. Jej cieľ je preskúmať, ako docieľiť efektívny a spoľahlivý prenos pomocou chýb odhaľujúcich algoritmov. Preskúmané algoritmy budú parita, kontrolný súčet, Hammingov kód a cyklický redundantný súčet. Takisto bude rozobraté, ako je možné algoritmy implementovať softvérovo respektíve hardvérovo. V nasledujúcom texte je platnou správou myslená správa, ktorá prešla overením po prijatí a správnou správou je myslená správa, ktorá bola odoslaná.

Jednosmerný prenos informácií je taký prenos, pri ktorom nie je možné dať späť vedieť odosielateľovi, že informácia ktorú odoslal, prišla poškodená, alebo je neaktuálna a vyžadovať znovuodoslanie. Tento prenos je používaný buď v prípadoch kedy nezáleží, ak prijaté dáta prišli poškodené, pretože sa ihneď stávajú neaktuálne. Takýto druh prenosu je napríklad živý prenos zvuku a obrazu, alebo je nerealizovateľné komunikovať späť. To môže byť napríklad pri prijímaní dát o počasí z družíc. Takýmto družicami sú napríklad NOAA15 alebo METEOR-M2 [10]. Tiež to môže byť kombinácia predchádzajúcich možností. Napríklad GPS signál, kedy je pre mobil nemožné komunikovať späť a zároveň sa informácie o polohe zariadenia stávajú neaktuálne.

Sériový prenos informácií je prenos, pri ktorom sú bity posielané za sebou na rozdiel od paralelného prenosu, kedy sú súčasne prenesené všetky bity. Jasnou výhodou sériového prenosu je jeden vodič, poprípade úzke pásmo, avšak za cenu rýchlosti prenosu. Preto sa sériovo komunikuje na dlhé vzdialenosti. Sériový prenos sa delí na synchronný a asynchronný. Synchronný prenos má navyše jeden signál, ktorým hovorí prijímaču v ktorý moment si má odoberať prichádzajúce bity. Ak ide o asynchronný prenos, prijímač aj vysielač sa musia synchronizovať a poznať časové rozostupy medzi bitmi [11].

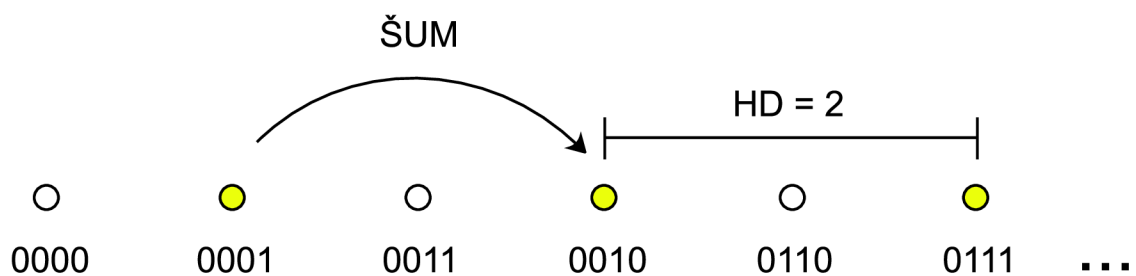
Dáta môžu byť prenášané medzi dvoma miestami, alebo aj v čase (zápisom do úložiska a neskorším čítaním). V niektorých prípadoch sa môže stávať, že vyčítané dáta sa budú líšiť od pôvodných. V ceste prenosu sa totiž môžu nachádzať zdroje rušenia, ktoré pozmenili časti dát. Poškrabané CD, presluchy vo vedení a pod. V takýchto prípadoch musí byť jasné, či prijaté, respektíve prečítané dáta sú platné, alebo najst' chybu a opraviť ju. Ak má byť zrealizovaný spoľahlivý systém na prenos dát, je potrebné ošetriť dve oblasti. Tou prvou je znížiť rušivé vplyvy. Napríklad posunom do iného pásma, alebo

zlepšením filtrácie a pod. Druhou oblasťou je detekcia chýb. Tá najčastejšie funguje tak, že sa na koniec posielaných dát pridá časť, ktorá overuje ich platnosť. Táto časť sa nazýva redundancia. Aj keď je vo väčšine prípadov na konci, sú systémy v ktorých je na začiatku, alebo rozdelená na viac častí v dátach. Je dôležité si uvedomiť, že sa nikdy nedokáže docieľiť stopercentná spoľahlivosť odhalenia chyby. Môže nastať situácia, pri ktorej rušenie zmení platné dáta na iné, ktoré sú taktiež platné. Najjednoduchší príklad je pri prenose s overením parity, kedy pri zmene párneho počtu bitov sa dáta znovu stávajú platné, aj keď už nebudú správne. Riešením je zmena algoritmu za taký, ktorý zväčší oblasť nesprávnych kombinácií dát.

Rôzne algoritmy majú rôznu efektivitu, čo je pomer medzi dátovými bitmi a všetkými posielanými bitmi. Napríklad obyčajná UART komunikácia posiela jeden bajt s jedným paritným bitom, z čoho je možné vypočítať efektivitu

$$\eta = \frac{N_D}{N_T} \cdot 100 \% = \frac{8}{9} \cdot 100 \% \approx 89 \%, \quad (2.1)$$

kde N_D je počet dátových bitov a N_T je počet všetkých bitov. Efektivita je teda 89 % s redundanciou 1 bit. Druhý dôležitý parameter overovacích algoritmov je Hammingova vzdialenosť (z angličtiny „*Hamming Distance*“, skrátene HD). Je to minimálny počet bitov, ktoré je možné zmeniť a stále nezaznamenať chybu. Napríklad spomínaná parita má $HD = 2$. Na obrázku 2.1 je naznačená predstava HD pre paritu. Ak sa zoradia všetky kombinácie ľubovoľnej správy v Grayovom kóde (v ktorom sa mení vždy iba jeden bit). HD bude minimálna vzdialenosť medzi dvoma platnými správami (označené žltou), z ktorých bola správna iba jedna. Je vhodné preto dopredu analyzovať systém a zistiť ako často dochádza k náhodným chybám. Potom určiť algoritmus a objem dát, ktorý bude overovať tak, aby šanca neodhalenej chyby bola pod chcené minimum.



Obrázok 2.1 Príklad platných správ overených paritou

2.1 Parita

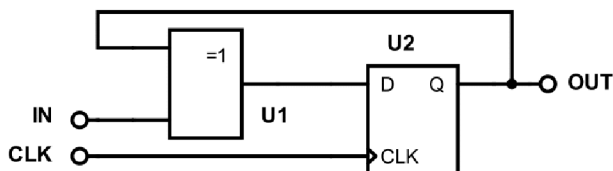
Parita je najjednoduchší zo spomínaných algoritmov. Pre paritu je $HD = 2$. Systémy s nízkym šumom využívajú paritu pomerne často (napríklad komunikácia medzi čipmi na jednej DPS). V týchto prípadoch je paritný bit za jedným bajtom čo absolútne postačuje, pretože šum v týchto obvodoch takmer neexistuje. Parita sa efektívne programovo počíta funkciou \oplus (XOR). Paritný bit 4 bitov bude

$$b_p = b_0 \oplus b_1 \oplus b_2 \oplus b_3. \quad (2.2)$$

Tak isto sa môže počítať z náhľadovej tabuľky, alebo pri C/C++ programovaní použiť vstavanú GCC funkciu

$$_builtin_parity(unsigned int x), \quad (2.3)$$

ktorá vypočíta paritu z počtu jednotiek modulo 2. V hardvéri sa môže počítať viacerými spôsobmi. Pri sériovom prenose sa používa zapojenie na obrázku 2.2. Klopny obvod D drží prechádzajúcu hodnotu parity a XOR vypočíta novú paritu pri príchode bitu.



Obrázok 2.2 Hardvérová kalkulačka parity

2.2 Kontrolný súčet

Kontrolný súčet (z angličtiny „*check sum*“) je o trochu robustnejší spôsob zabezpečenia dát, avšak nie o veľa. Kontrolný súčet má $HD = 2$, rovnako ako parita, keďže pri zmene 2 „správnych“ bitov už nie je možné chybu rozoznať. V čom je robustnejší oproti parite je, že to nemôžu byť náhodné bity, ale bity s rovnakou pozíciou (rovnakým exponentom), pretože v súčte má každý bit inú váhu, od najväčšej (MSB), po najmenšiu (LSB).

Existuje viacero spôsobov výpočtu kontrolného súčtu. Ak sa spraví obyčajný súčet jednotlivých bajtov, poprípade 16 bitových slov, alebo inak dlhých nedeliteľných celkov správy. Narazí sa na problém, že výsledný súčet je o niečo dlhší ako prenášaný celok. Napríklad je možné si predstaviť protokol, ktorý prenáša 16 bajtov po sebe. Je

jasné, že súčet všetkých bajtov sa nezmesí do jedného bajtu. Preto sa tento problém rieši použitím dvoch bajtov na výsledok kontrolného súčtu. Popríklad odrezaním pretečenej časti. Alebo treťou možnosťou, odrezaním pretečenej časti a jej znovu pričítaním k zvyšku. Posledný spôsob využíva aj internetový protokol, s tým že výsledná hodnota je ešte invertovaná a posiela sa na konci správy [22]. Jedná sa teda o jednotkový doplnok. Prijímač potom sčíta všetky prijaté nedeliteľné celky (aj vrátane redundancie) a takisto pretečenú časť redundancie odreže a znovu pripočíta (v prípade internetového protokolu sú to 16 bitové slová). Pre platnú správu bude výsledok poslednej operácie sčítania maximálne 16 bitové číslo 0xFFFF. Takže ak by správa obsahovala 3 slová s hodnotou 0xFF15, ako prvý krok je ich sčítanie

$$\begin{array}{r}
 F \quad F \quad 1 \quad 5 \\
 F \quad F \quad 1 \quad 5 \\
 + \quad F \quad F \quad 1 \quad 5 \\
 \hline
 2 \quad F \quad D \quad 3 \quad F
 \end{array} , \tag{2.4}$$

číslo 2 pretieklo a teda sa znovu pripočíta

$$\begin{array}{r}
 F \quad D \quad 3 \quad F \\
 + \quad \quad \quad \quad 2 \\
 \hline
 F \quad D \quad 4 \quad 1
 \end{array} . \tag{2.5}$$

Bitovou inverziou sa získa požadovaná redundancia, ktorá sa odošle na konci s ostatnými číslami

$$\sim FD41 = 02BE. \tag{2.6}$$

Prijímač potom opakuje rovnaké kroky aj s redundanciou

$$\begin{array}{r}
 F \quad F \quad 1 \quad 5 \\
 F \quad F \quad 1 \quad 5 \\
 F \quad F \quad 1 \quad 5 \\
 + \quad 0 \quad 2 \quad B \quad E \\
 \hline
 2 \quad F \quad F \quad F \quad D
 \end{array} . \tag{2.7}$$

Dáta sú teda platné, pretože výsledok je 0xFFFF, respektíve po bitovej inverzii 0

$$\begin{array}{r}
 F \quad F \quad F \quad D \\
 + \quad \quad \quad \quad 2 \\
 \hline
 F \quad F \quad F \quad F
 \end{array} . \tag{2.8}$$

2.3 Hammingov kontrolný súčet

Tento algoritmus generuje paritu pre množiny bitov v správe. Konkrétne množiny bitov pre 16 bitový celok sú v tabuľkách 2.1 až 2.4. Jednotlivé množiny sa prekrývajú, z čoho je možné odhaliť pozíciu chybného bitu, ak sa v správe nejaký nachádza. Ak sú v správe 2 chybné bity, algoritmus to odhalí, avšak nedokáže zistiť ktoré to sú. Hammingov kód má $HD = 3$ a bol to jeden z prvých algoritmov na odhaľovanie chýb. Tento algoritmus vynášiel Richard Wesley Hamming v roku 1950 [25]. To, v čom je tento algoritmus výnimočný, je jednoduchosť jeho softvérovej implementácie.

Ak je jeden celok správy 16 bitov, je ho možné uložiť do tabuľky 4×4 bity. V tabuľkách sú zaznačené binárne indexy bitov od 0000 (0) po 1111 (15). Takisto sú v tabuľkách farebne odlišené miesta, kde index na prvom, druhom, treťom a štvrtom bite má hodnotu 1 (spomínané množiny). Pôvodný Hammingov kód počíta paritu pre tieto množiny s tým, že výsledok každej ukladá na miesto označené najtmavšou farbou. Ich indexy sú decimálne {1;2;4;8}. Tieto 4 paritné bity zaberajú miesto v správe, navyše nie je možné použiť ani nultý bit označený X (prečo bude vysvetlené neskôr), takže zo 16 prenesených bitov je 11 užitočných, ktorých indexy sú decimálne {3;5;6;7;9;10;11;12;13;14;15}. Jediné čo je potrebné pre získanie štyroch paritných bitov je XOR všetkých indexov, ktoré v sebe držia 1. Ak by 11 prenášaných bitov bolo {1;0;1;0;0;1;0;0;1;0;0}, potom 4 paritné bity by boli

$$par = 3 \oplus 6 \oplus 10 \oplus 13 = 2, \quad (2.9)$$

takže binárne 0010. Viac zrejme to je pri binárnom zápise

$$\begin{array}{cccc} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ \oplus & 1 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 \end{array}, \quad (2.10)$$

kde každý stĺpec počíta paritu pre jednu z množín. Paritné bity musia byť pred výpočtom samozrejme 0. Vysielač vyplní paritné bity a správu odošle. Prijímač vykoná rovnakú operáciu, ale teraz v nej budú zahrnuté indexy paritných bitov, ktoré obsahujú 1 (v tomto prípade jediný index 2, ostatné 1,4 a 8 sú nulové)

$$\begin{array}{rcccc}
 & 0 & 0 & 1 & 1 \\
 & 0 & 1 & 1 & 0 \\
 & 1 & 0 & 1 & 0 \\
 & 1 & 1 & 0 & 1 \\
 \oplus & 0 & 0 & 1 & 0 \\
 \hline
 & 0 & 0 & 0 & 0
 \end{array}
 \quad (2.11)$$

Pri výsledku 0 prijímač vie, že dáta sú platné. Ak by sa prenosom zmenil náhodný bit z 0 na 1, jeho index bude zahrnutý pri operácii XOR a teda výsledok nebude nula, ale práve index zmeneného bitu. Toto je unikátne na Hammingovom algoritme, ktorý takto ukáže na chybný bit a teda je ho možné opraviť. To isté platí, aj ak by bol zmenený náhodný bit z 1 na 0. Pretože index toho bitu nebude zahrnutý v XOR operácii, zmení sa (respektíve otočí) výsledok parity v množine, v ktorej bit logickej jednotky chýba. Čo vo výsledku ukáže na chybný bit [23]. Problém prenosu na nultom bite je, že ho neobsahuje žiadna množina a ak sa zmení, nie je to možné rozoznať. Preto sa tento bit používa na paritu celej správy pre extra ochranu dát. Takémuto algoritmu sa hovorí rozšírený Hammingov kód.

Tabuľka 2.1 Index s 1 na 4. mieste

X	0	1	1
0000	0001	0010	0011
0	0	1	0
0100	0101	0110	0111
0	0	1	0
1000	1001	1010	1011
0	1	0	0
1100	1101	1110	1111

Tabuľka 2.3 Index s 1 na 3. mieste

X	0	1	1
0000	0001	0010	0011
0	0	1	0
0100	0101	0110	0111
0	0	1	0
1000	1001	1010	1011
0	1	0	0
1100	1101	1110	1111

Tabuľka 2.2 Index s 1 na 2. mieste

X	0	1	1
0000	0001	0010	0011
0	0	1	0
0100	0101	0110	0111
0	0	1	0
1000	1001	1010	1011
0	1	0	0
1100	1101	1110	1111

Tabuľka 2.4 Index s 1 na 1. mieste

X	0	1	1
0000	0001	0010	0011
0	0	1	0
0100	0101	0110	0111
0	0	1	0
1000	1001	1010	1011
0	1	0	0
1100	1101	1110	1111

Čím väčšia správa, tým je efektívnosť algoritmu väčšia, pretože za každé zdvojnásobenie dát pribudne jeden paritný bit. Avšak nie je dobré mať príliš veľa dát overených naraz. Pri väčšom objeme dát je síce efektivita algoritmu väčšia, ale tak isto aj pravdepodobnosť výskytu chyby. Dobré riešenie je najskôr preskúmať prenosovú sústavu a na základe toho určiť maximálnu dĺžku správy tak, aby sa v nej nevyskytlo viac chýb, ako je možné rozoznať.

2.3.1 Implementácia do softvéru

Problém trochu nastáva pri delení dát. Spomínaný 16 bitový blok potrebuje na paritu jednotlivých množín 4 bity, 11 ostáva pre dáta a nulté miesto sa nedá použiť. Pri 32 bitovom bloku ich je potrebných pre paritu 5, 26 ostane pre dáta a nultý sa nedá použiť a tak ďalej. Otázkou je, ako rozdeliť správu (poprípade ju nedeliť, ak je dostatočne malá) do voľných miest pre dáta. Keďže správa je takmer vždy násobkom 8 bitov. Riešením je použiť 2-krát väčšiu tabuľku, ako dáta potrebujú. Posunúť všetky dátové bity indexovo vyššie o jedna (aby sa nemusela použiť nultá pozícia) a vypočítanú paritu pre množiny nedávať na miesta v tabuľke, ale odoslať ako samostatné dáta za správou (pretože tieto miesta zaberie správa). Aj keď už nepôjde o originálny Hammingov kód, podstata ostane rovnaká. Netreba sa nechať pomýliť, neznamená to že sa bude prenášať 2-krát toľko bitov. Tabuľka je len pomôcka na pochopenie algoritmu. Dôležité sú indexy bitov, ktoré sa tým zmenia. Takže ak by správa mala mať 16 bitov, vloží sa do 32 bitovej tabuľky (4 x 8). Prvý bit správy pôjde na pozíciu s indexom 00001 (decimálne 1). Je potrebné si všimnúť, že index je v tomto prípade 5 bitové číslo. Posledný bit pôjde na pozíciu 10000 (decimálne 16, preto 2x väčšia tabuľka). Jediná nevýhoda je, že parita jednotlivých množín (redundancia) bude v tomto prípade o jeden bit dlhšia. Toto paritné číslo sa vypočíta rovnako, ako XOR všetkých indexov na ktorých je bit logická jednotka. S ostatnými bitmi v pomyselnnej tabuľke 4 x 8 sa ani nepočíta, ani sa neprenášajú. Ako bolo povedané, ide len o pomôcku pre predstavu pri pridelovaní indexov.

Nasledujúci názorný kód vygeneruje náhodných 16 bitov, ku ktorým vypočíta Hammingov kód s modifikáciou vysvetlenou v odstavci vyššie. Kód je napísaný v JavaSkripte. Bit na 7. mieste bol zmenený a simuluje prenosovú chybu. Prijímač vykoná XOR so všetkými indexami s bitom logickej jednotky a nakoniec aj s paritným číslom vygenerovaným vysielačom, ktoré tak isto prijal. Výsledok tejto operácie je index, na ktorom bol bit zmenený, poprípade 0, ak sa dáta nezmenili. V tomto prípade sa vypíše

„chyba na pozícii 7“. Samozrejme ráta sa s tým, že pri prenose nevznikne väčší počet chýb ako 2, pretože HD tohto algoritmu je 3 a Hammingov kód už neodhalí 3 „správne“ zmenené bity.

```
//vysielač
var arr = []
//generovanie náhodných 16 bitov
for (var i = 0; i < 16; i++) {
  arr.push({ index: i + 1, value: Math.round(Math.random()) })
}
//generovanie redundancie
var xorResultTransmitter
arr.forEach(element => {
  if (element.value) {
    xorResultTransmitter ^= element.index
  }
})

//prenos
arr[7].value = !arr[7].value //simulovaná chyba

//prijímač
var xorResultReceiver
//výpočet vlastnej redundancie
arr.forEach(element => {
  if (element.value) {
    xorResultReceiver ^= element.index;
  }
})
//porovnanie s prijatou redundanciou
xorResultReceiver ^= xorResultTransmitter
if(xorResultReceiver != 0){
  //upozornenie o chybe
  console.log(`chyba na pozícii ${xorResultReceiver - 1}`)
}
```

2.4 Cyklický redundantný súčet

Cyklický redundantný súčet (skratka CRC) je z predošlých algoritmov najodolnejší, ale je aj najťažší na pochopenie. Využíva ho napríklad Bluetooth, Ethernet alebo .png formát [30]. Vynašiel ho v roku 1961 Wesley Patterson [31]. Tento algoritmus považuje dáta za jeden celok, konkrétne polynóm. Tento polynóm sa delí druhým (deliacim) polynómom a výsledok sa vkladá na koniec posielanej správy. Algoritmus vychádza z toho, že berie

celú správu ako celok, ktorý doplní o takú časť, aby bola deliteľná dohodnutým výrazom. Pre príklad (zatiaľ nie CRC), správa „Hi!“ má binárnu reprezentáciu

$$ASCII("Hi!") = 0b\ 01001000\ 01101001\ 00100001. \quad (2.12)$$

Toto číslo sa posunie doľava o 8, aby ostalo vpravo miesto na zvyšok po delení. V tomto prípade sa bude deliť napríklad prvočíslom 251. Pripravená správa na výpočet takejto redundancie je binárne

$$m = 01001000\ 01101001\ 00100001\ 00000000, \quad (2.13)$$

alebo decimálne

$$m_{DEC} = 1214849280. \quad (2.14)$$

Po celočíselnom delení čísla 1214849280 číslom 251 ostane zvyšok 244. Takže ak sa vloží na vytvorené prázdne miesta 251 – 244, čo je 7 (00000111 binárne), bude celá správa deliteľná 251. Prijímač prijatú správu vydolí rovnakým číslom a pre nepoškodenú správu očakáva nulový zvyšok. Výsledná odoslaná správa by bola

$$t = 01001000\ 01101001\ 00100001\ 00000111. \quad (2.15)$$

Problém tohto príkladu spočíva v interpretácii chyby. Ak sa v správe zmenia napríklad 2 bity označené červenou a prijatá správa bude

$$r = 01001000\ 01101111\ 00100001\ 00000111. \quad (2.16)$$

Z matematického hľadiska prijímač prijal správu r

$$r = t + e, \quad (2.17)$$

kde t je odoslaná správa a e je chyba, ktorá sa pridala pri prenose. V tomto prípade to je jednoduché. Chyba je

$$e = 00000000\ 00000110\ 00000000\ 00000000, \quad (2.18)$$

pretože správa t je deliteľná 251. Pokiaľ bude chyba e nedeliteľná 251, bude chyba detekovateľná. Čo ale ak prijatá správa bude

$$t = 01001000\ 00001001\ 00100001\ 00000111, \quad (2.19)$$

teraz je chyba namiesto 2 bitov diferencially rozdielna

$$e = 11111111 10100000 00000000 00000000. \quad (2.20)$$

Aj keď ide o rovnakú dvojbitovú chybu, len v opačnom smere, teraz nedáva zmysel určovať či je deliteľná 251 a je ťažké matematicky určiť, či je takáto chyba odhaliteľná. Toto je problém prenosu z nižšieho rádu pri sčítaní (z angličtiny "carrying"). Ten CRC algoritmus obchádza, aby vedel matematicky vyjadriť, aké chyby je možné odchytiť a aké nie. Preto CRC nepočíta v množine reálnych čísel ale iba v konečnej množine dvoch čísel 0 a 1.

S algebrou je možné počítať aj v inej množine ako je množina reálnych čísel, napríklad v množine komplexných čísel a tak isto v konečnej množine reálnych čísel. Ak má algebra pre množinu fungovať, je potrebné aby sa množina pri počítaní riadila konkrétnymi pravidlami.

Tabuľka 2.5 Algebraické axiómy pre množiny [35]

pravidlo	sčítanie	násobenie
asociativita	$(a + b) + c = a + (b + c)$	$(a \cdot b) \cdot c = a \cdot (b \cdot c)$
komutativita	$a + b = b + a$	$a \cdot b = b \cdot a$
distributivita	$a \cdot (b + c) = a \cdot b + a \cdot c$	$(a + b) \cdot c = a \cdot c + b \cdot c$
identita	$a + 0 = a$	$a \cdot 1 = a$
inverzia	$a + (-a) = 0$	$a \cdot \frac{1}{a} = 1$

Axiómy z tabuľky 2.5 spĺňa aj binárna množina reálnych čísel $\{0;1\}$. To je dôležité, lebo algebra je potrebná na delenie polynómov. Konečná binárna množina bude zadaná nasledovne [33]

$$\begin{array}{cccc}
 0 + 0 = 0 & 0 + 1 = 1 & 1 + 0 = 1 & 1 + 1 = 0 \\
 0 - 0 = 0 & 0 - 1 = 1 & 1 - 0 = 1 & 1 - 1 = 0 \\
 0 \cdot 0 = 0 & 0 \cdot 1 = 0 & 1 \cdot 0 = 0 & 1 \cdot 1 = 0 \\
 0/1 = 0 & 1/1 = 1. & &
 \end{array} \quad (2.21)$$

Aj keď sa to javí zvláštne (napríklad $1 + 1 = 0$), tak s takto zadanou množinou sú axiómy splnené. Je tu však ešte jeden problém. Ak sa teda počíta v konečnej množine čísel s najvyšším číslom 1, ako je možné vyjadriť celú správu jedným veľkým číslom? Toto sa vyrieši tým, že sa správa vyjadří ako polynóm. Takto sa s ním dajú vykonávať matematické úkony a to zároveň v konečnej množine. Takže pripravená správa pre výpočet CRC (s nulami na konci) „Hi!“ už nebude (2.13), ale

$$m(x) = x^{30} + x^{27} + x^{22} + x^{21} + x^{19} + x^{16} + x^{13} + x^8. \quad (2.22)$$

Takisto, namiesto reálneho čísla sa správa delí polynómom. V tomto prípade 9 bitovým CRC-8 (8 stupňovým polynómom) $0x107$ (prvý bit sa pri HEX zápise niekedy neuvádza a preto v niektorej literatúre môže byť uvedený ako $0x07$). Deliaci polynóm bude označený ako dp

$$dp(x) = x^8 + x^2 + x^1 + 1. \quad (2.23)$$

Pretože x^0 je 1, posledné číslo polynómu bude iba 1. Delenie je na obrázku 2.3. Prebieha rovnako ako delenie polynómov s obmedzením konečnej množiny čísel.

$+x^8 + x^2 + x^1 + 1$							Deliteľ
$+x^{22} + x^{19} + x^{16} + x^{15}$	$+x^{12}$	$+x^{10}$	$+x^8 + x^7 + x^6$	$+x^3$			Podiel
$+x^{30} + x^{27}$	$+x^{22} + x^{21}$	$+x^{19}$	$+x^{16}$	$+x^{13}$	$+x^8$	Delenec	
$+x^{30}$	$+x^{24} + x^{23} + x^{22}$						
$+x^{27}$	$+x^{24} + x^{23}$	$+x^{21} + x^{19}$	$+x^{16}$	$+x^{13}$	$+x^8$		
$+x^{27}$	$+x^{21} + x^{20} + x^{19}$						
$+x^{24}$	$+x^{23}$	$+x^{20}$	$+x^{16}$	$+x^{13}$	$+x^8$		
$+x^{24}$			$+x^{18} + x^{17} + x^{16}$				
$+x^{23}$	$+x^{20}$	$+x^{18} + x^{17}$		$+x^{13}$	$+x^8$		
$+x^{23}$		$+x^{17} + x^{16} + x^{15}$					
	$+x^{20}$	$+x^{18}$	$+x^{16} + x^{15}$	$+x^{13}$	$+x^8$		
	$+x^{20}$		$+x^{14} + x^{13} + x^{12}$				
		$+x^{18}$	$+x^{16} + x^{15} + x^{14}$	$+x^{12}$	$+x^8$		
		$+x^{18}$		$+x^{12} + x^{11} + x^{10}$			
			$+x^{16} + x^{15} + x^{14}$	$+x^{11} + x^{10}$	$+x^8$		
			$+x^{16}$	$+x^{10} + x^9 + x^8$			
			$+x^{15} + x^{14}$	$+x^{11}$	$+x^9$		
			$+x^{15}$		$+x^9 + x^8 + x^7$		
			$+x^{14}$	$+x^{11}$	$+x^8 + x^7$		
			$+x^{14}$		$+x^8 + x^7 + x^6$		
				$+x^{11}$	x^6		
				$+x^{11}$		$+x^5 + x^4 + x^3$	
						$+x^6 + x^5 + x^4 + x^3$	Zvyšok (CRC)

Obrázok 2.3 Ručný výpočet CRC-8

Keďže sa počítalo v konečnej množine, koeficienty pri jednotlivých exponentoch vyšli iba 0 a 1. Je to kvôli tomu, že sčítanie a odčítanie sú rovnaké operácie. Zvyšok po delení je výsledná hodnota CRC, ktorá sa doplní na voľné miesta v pripravenej správe, konkrétne odpočítaním od pripravenej správy. Odosielaná správa bude

$$t(x) = x^{30} + x^{27} + x^{22} + x^{21} + x^{19} + x^{16} + x^{13} + x^8 + x^6 + x^5 + x^4 + x^3, \quad (2.24)$$

alebo binárne

$$t = 01001000\ 01101001\ 00100001\ 01111000. \quad (2.25)$$

Zvyšok po delení sa nemusí odpočítavať od polynómu a potom pripočítavať k správe. Môže sa priamo odpočítavať od správy, čím sa stane správa deliteľná polynómom. Pretože v binárnej konečnej množine neexistuje číslo -1, po odčítaní zvyšku od pripravenej správy $m(x)$ bude správa vyzerat', ako by sa k nej zvyšok pričítal. V skutočnosti sa ale odčítal. Chyba teraz netrpí prenosom z nižšieho rádu pri sčítaní, ale vyjadruje, na ktorých pozíciách sa otočil bit. Ak bude prijatá správa bez dvoch bitov logických jednotiek, tak ako predtým

$$r(x) = x^{30} + x^{27} + x^{19} + x^{16} + x^{13} + x^8 + x^6 + x^5 + x^4 + x^3, \quad (2.26)$$

chyba bude mať iba 2 členy

$$e(x) = x^{22} + x^{21}. \quad (2.27)$$

Teraz je možné zistiť a matematicky dokázať aké chyby algoritmus odhalí. Pretože, pokiaľ bude chyba nedeliteľná polynómom, prijímaču ostane nenulový zvyšok a chybu odhalí. Podľa toho sa aj volí polynóm, ktorým sa CRC počíta. Najskôr sa meria a zaznamená, aké chyby sa vyskytujú v prenosovej sústave a na základe toho sa vyberie polynóm. Podmienkou je, aby nebol násobkom zaznamenaných chýb. Napríklad, ak sa pri prenose vyskytujú 1 bitové chyby

$$e(x) = x^i, \quad (2.28)$$

kde i je pozícia chyby v správe. Polynóm, ktorý nie je celočíselným násobkom x^i je každý polynóm, ktorý má viac ako jeden člen, napríklad

$$dp(x) = x + 1. \quad (2.29)$$

Preto každý polynóm s viac ako jedným členom má $HD = 2$. Pre dvojbitové chyby sa chyba všeobecne vyjadrí ako

$$e(x) = x^i + x^j; \quad i < j. \quad (2.30)$$

Chyba sa upraví do tvaru viacerých činiteľov a volí sa polynóm, ktorý nie je násobkom ani jedného z činiteľov, takže

$$\begin{aligned} e(x) &= x^i \cdot (x^{i-j} + 1), \\ e(x) &= x^i \cdot (x^k + 1); \quad k = i - j. \end{aligned} \quad (2.31)$$

Číslo k je vzdialenosť medzi dvoma chybami. Prvý činiteľ x^i (rovnako ako v prvom prípade) nebude násobkom, ak má polynóm viac ako 2 členy. Druhý činiteľ $(x^k + 1)$ nebude násobkom, ak k bude menšie ako určité číslo. Napríklad v prípade teraz použitého polynómu to je 96 bitov. Takže ak bude celková správa menšia ako 96, je zaručené, že polynóm $0x107$ bude mať $HD = 3$. Matematicky dokázať aká môže byť maximálne dlhá správa pre konkrétny polynóm a konkrétnu Hammingovu vzdialenosť je pomerne náročné. Najjednoduchšia cesta zistenia HD pre polynómy nižších rádov je hrubá sila (z angličtiny „*Brute-force*“). Profesor Philip Koopman z Carnegie Mellon University udržiava webstránku so svojim výskumom a novými polynómami. Na stránke je aj odkaz k stiahnutiu programu, ktorý hrubou silou vypočíta HD pre zadaný polynóm¹. Aj keď sa dnes prichádza na lepšie polynómy, ktoré dosahujú väčšie HD pre rovnako dlhé správy, niektoré sú už zaužívané ako štandardy a stále sa používajú, aj keď existujú lepšie náhrady [33].

Časté chyby v prenosových sústavách sú takzvané zhlukové chyby (z angličtiny „*burst errors*“). Je to poškodená časť správy s dĺžkou k , ktorá začína prvým poškodeným bitom a končí posledným poškodeným bitom. Vo vnútri medzi prvým a posledným poškodeným bitom sú náhodne niektoré bity poškodené a niektoré nie. Napríklad, môže sa stať chyba, ktorá spôsobí, že zmení všetky bity v prvom bajte na jednotky. Dĺžka zhluku je tým pádom 8 a prijatá správa by vyzerala

$$r = \mathbf{11111111} 01101001 00100001 01111000. \quad (2.32)$$

¹ [Odkaz na Philip Koopman, Best CRC Polynomials](#)

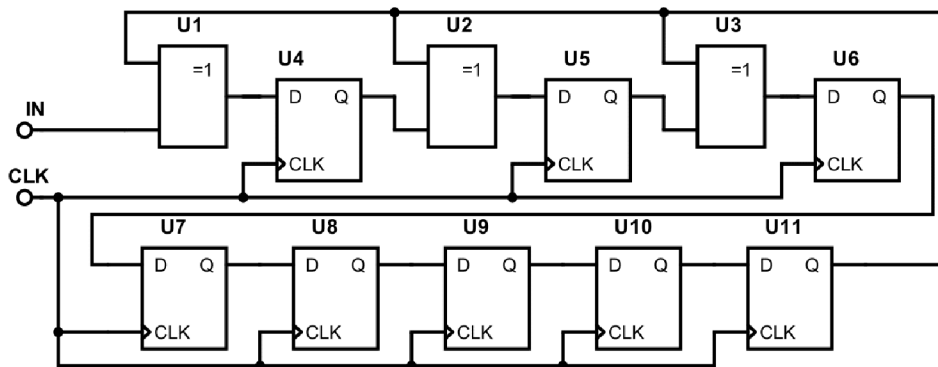
Červenou sú označené chybné bity. CRC má tú vlastnosť, že dokáže dobre odhaľovať tento druh chýb. Ak sa takýto druh chyby všeobecne vyjadrí súčinom

$$e(x) = x^i \cdot (x^{k-1} + x^{k-2} + \dots + 1). \quad (2.33)$$

Je zaručené, že chyba bude odhalená, ak deliaci polynóm bude mať viac ako jeden člen (podmienka pre prvý činiteľ) a zároveň jeho stupeň bude väčší ako stupeň polynómu druhého činiteľa. Teda 8 bitová zhluková chyba bude zachytená 8 stupňovým deliacim polynómom, 16 bitová, 16 stupňovým atď. [33].

2.4.1 Implementácia do hardvéru

Výhodou tohto algoritmu je aj jeho jednoduchosť implementácie do hardvéru. Pri pohľade na príklad delenia 2.3, je možné si všimnúť určitý opakujúci sa vzor. Deliaci polynóm je ôsmeho rádu a jeho členy sú 8,2,1 a 0. To čo je dôležité v tomto prípade, sú vzdialenosti ostatných členov od ôsmeho člena. Aj keď sa polynóm násobí vždy inou mocninou x , jeho vzdialenosti od tohto člena sa nemenia. Takto deliaci polynóm postupuje po celej správe a vždy, keď narazí na prvý člen správy (v ručne vypočítanom príklade to boli členy 30,27,24,23...), otočí bity správy na miestach vždy rovnako vzdialených od začiatku a „zahodí“ prvý člen správy [34]. V hardvéri namiesto chodenia s deliacim polynómom cez správu, bude správa prechádzať cez deliaci polynóm. Konkrétne sa správa bude posúvať cez posuvný register. Na XOR sa dá pozeráť ako na prepínateľný invertor/buffer. Takže na miesta správne vzdialené od prvého člena sa vloží XOR operand, pretože bity sa otáčajú iba v prípade, keď deliaci polynóm narazí na prvý člen (teda prvý člen s koeficientom 1, čiže logickú jednotku). V ostatnom prípade sa správa posúva bez zmeny. Bit, ktorý sa zahadzuje bude zodpovedný za riadenie XOR operandov. Hardvérový obvod na počítanie CRC-8 polynómu 0x107 je na obrázku 2.4.



Obrázok 2.4 Hardvérová kalkulačka na CRC-8, 0x107

Správa prichádza zľava spolu s hodinovým signálom. Po vsunutí celej správy ostane vypočítané CRC na jednotlivých D klopných obvodoch. Táto hodnota sa odošle za správou. Prijímaču pri platnej prijatej správe ostane po výpočte CRC na každom klopnom obvode 0.

2.4.2 CRC modifikácie

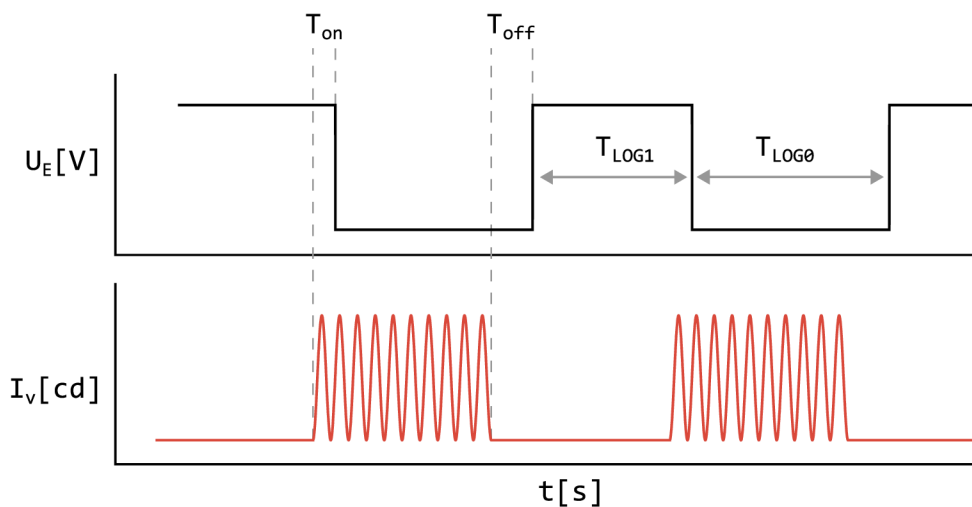
Okrem voľby deliaceho polynómu existujú aj viaceré modifikácie, ktoré robia CRC ešte robustnejším, popripade lepším pre konkrétne aplikácie. Tieto modifikácie sú

- „*Init*“,
- „*RefIn*“,
- „*RefOut*“,
- „*XorOut*“.

Hodnota „*Init*“ je inicializačná hodnota D klopných obvodov pred posúvaním správy. Ak by totižto z nejakého dôvodu boli dáta odpojené po celý čas výpočtu CRC na strane prijímača a prichádzal by iba hodinový signál. Ostala by v registri 0, čo je chyba, pretože správa sa neprijala. Je preto užitočné naplniť register napríklad jednotkami. V takomto prípade aj pri odpojených dátach neostane v registri 0. „*RefIn*“ (z angličtiny „*Reflect In*“) je nastavenie, ktoré invertuje všetky bity správy tak, ako sa posúvajú do vnútra registra. „*RefOut*“ (z angličtiny „*Reflect Out*“) na druhú stranu otočí všetky CRC bity po jeho vypočítaní. Posledný „*XorOut*“ vykoná XOR s vypočítaním CRC a zadaným číslom. Okrem toho je možné nájsť aj hodnotou „*Check*“ udávanú pri polynómoch, aj s ich konkrétnou modifikáciou. Tento údaj je vypočítaná CRC hodnota pre ASCII správu 123456789. Je to forma kontroly pre tých, ktorí s algoritmom robia, alebo ho implementujú.

3 NÁVRH ZARIADENIA

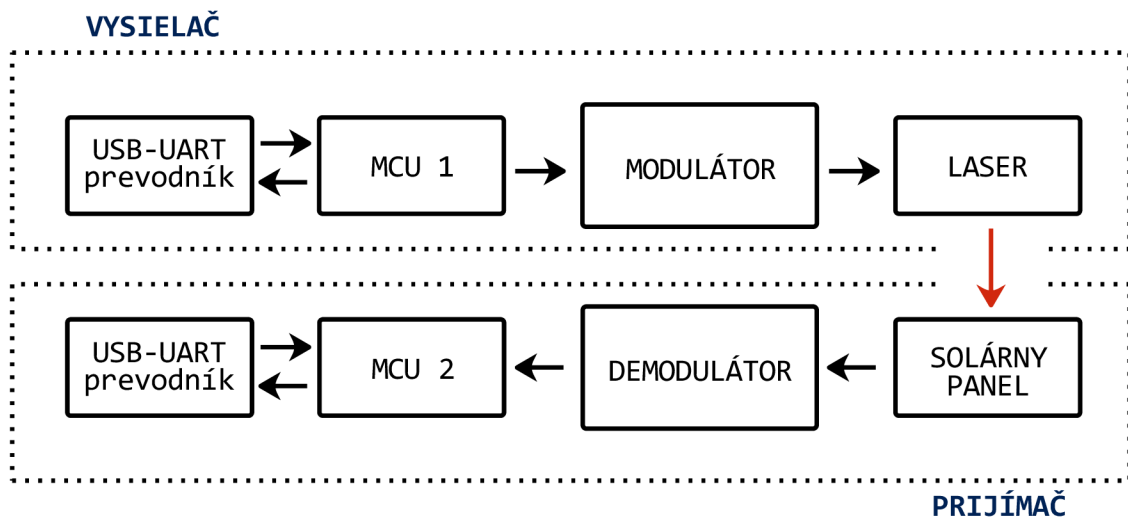
V tejto kapitole bude prebratý princíp činnosti a definovaný spôsob riešenia celého zariadenia. Na konci kapitoly budú zadané jednotlivé vrstvy a ich miesto implementácie. Pre zopakovanie, cieľom je realizovať systém pozostávajúci z vysielateľa a prijímateľa, ktorý bude prenášať dáta pomocou laserového lúča. Modulovať sa bude digitálny signál, na čo bude postačovať metóda ASK (z angličtiny „*Amplitude Shift Keying*”), kde je digitálny signál reprezentovaný prítomnosťou alebo neprítomnosťou nosnej vlny [15]. Je žiadúce realizovať systém s čo najvyšším kmitočtom nosného signálu, aby bolo možné prenášať aj väčšie súbory v čo najkratšom čase. Zároveň to pomôže k zrýchleniu merania náhodných chýb pri prenose. Na obrázku 3.1 je červenou intenzita laserového lúča a čiernou demodulovaný signál. Vyznačené sú aj oneskorenia, kvôli konečnej rýchlosti šírenia signálu a zotrvačnosti obvodu. Toto má za následok nerovnomerný pomer času medzi logickou jednotkou a logickou nulou, na čo si treba pri vzorkovaní dávať pozor.



Obrázok 3.1 Metóda kódovania ASK

Bloková schéma celého zariadenia je na obrázku 3.2. Vysielač aj prijímač sa pripoja k dvom rozdielnym počítačom cez USB. Po pripojení sa vytvorí virtuálny sériový port, cez ktorý bude možné komunikovať. Mikrokontrolér prijaté dáta z počítača zabezpečí redundanciou, vygenerovanou jedným z vybraných algoritmov a pošle do modulátora. Modulované dáta sa prenesú laserom na solárny panel. Na druhej strane sa demodulujú, druhý mikrokontrolér ich vyčíta a overí rovnakým algoritmom. Overené dáta pošle cez USB do druhého počítača. Ak by mikrokontrolér dáta nedokázal overiť, tak ich zahodí.

Aktívna úroveň logického signálu v hardvérovom dizajne bude logická nula. To znamená, že vo vysieláči logická nula zapne moduláciu a logická jednotka ju vypne. Tak isto v prijímači prečítanie logickej nuly znamená, že laserový lúč dopadá na solárny panel a prečítanie logickej jednotky, že laser je vypnutý. Takže napríklad ak bude chcieť užívateľ odoslať medzeru, ktorá má bitovú UTF-8 reprezentáciu 00100000, bude laser väčšinu času zapnutý.

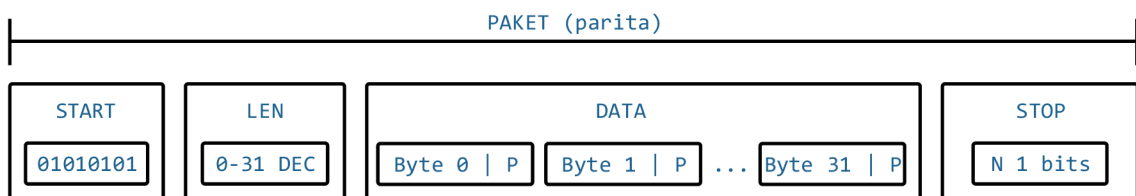


Obrázok 3.2 Bloková schéma celého systému

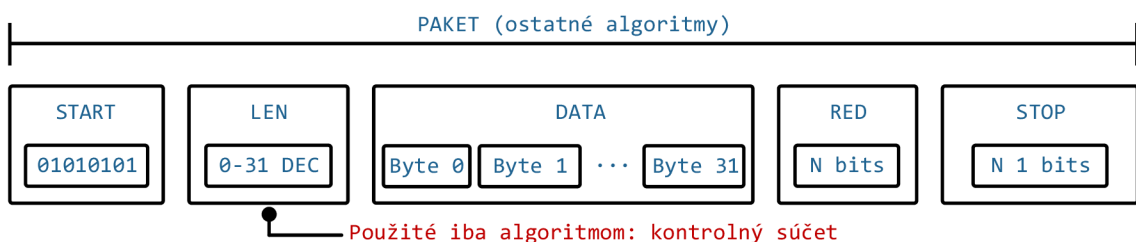
3.1 Schéma paketu

Prenos bude pozostávať z paketov. Každý paket bude začínať START sekvenciou bitov. Komunikácia UART má napríklad jeden START bit (logickú nulu na začiatku prenosu), ktorý indikuje jeho začiatok. V tomto prípade je situácia trochu komplikovanejšia, pretože rušenie je oveľa väčšie a k strate synchronizácie môže dôjsť oveľa ľahšie. Väčšina zdrojov rušenia bude typu svetelných zábleskov, ktoré zasiahnu solárny panel, alebo vo forme rádiového signálu, ktorý sa naindukujú na jeho vodičoch. Aj keď bude elektronika stienená, aktívnu časť solárneho panelu nie je možné zakryť. Okrem problému s rušením, je tu problém zablokovania lúča nepriehľadným predmetom. V oboch prípadoch môže dôjsť k strate synchronizácie. Pre potlačenie týchto problémov budú na začiatku každého paketu START bity s inou frekvenciou, ako je frekvencia prenosu dát. Konkrétne 8 párov 01. STOP bity budú obsahovať niekoľko log. jednotiek na oddelenie paketov, aby mal prijímač čas odoslať dáta do počítača. Koľko ich bude sa určí až neskôr, keď bude zariadenie pracovať. Prečo sú START a STOP bity v takejto forme je vysvetlené v nasledujúcej podkapitole Synchronizácia.

Pakety budú niest' maximálne 32 bajtov dát. Číslo 32 bolo zvolené pretože je dostatočne veľké pre algoritmy CRC a Hammingov kód a overovať kratšie dáta týmito algoritmami je neefektívne. Tieto dáta sa budú ďalej nazývať DATA bity/bajty popripade iba dáta. Ostatné algoritmy (parita a kontrolný súčet) nepotrebujú na prenos 32 bajtov dát. S nimi bude možné preniesť aj menší obsah tak, že po START bitoch bude nasledovať 5 bitov určujúcich počet DATA bajtov v pakete. Ich názov bude LEN bity z anglického slova length. Po prečítaní týchto bitov bude mikrokontrolér vedieť ako dlho vzorkovať. Takže premenná dĺžka dát bude iba pri algoritmoch parita a kontrolný súčet a LEN bity budú pri druhých dvoch algoritmoch vynechané. 5 LEN bitov vie dať 32 kombinácií, decimálne 0 až 31. Takže 0 bude znamenať že správa obsahuje všetkých 32 bajtov a ostatné kombinácie budú číselne odpovedať počtu bajtov v správe. Ako posledné ostávajú bity pre overenie platnosti správy. Ich funkcia je overovať dáta z paketu. Tieto bity sa budú značiť RED ako redundancia. Ich počet bude iný pre každý overovací algoritmus. Napríklad parita potrebuje 1 bit za každým DATA bajtom (preto bude mať parita inú štruktúru paketu ako ostatné algoritmy), zatiaľ čo 32 bitový cyklický redundantný súčet (CRC-32) potrebuje 32 bitov na konci. Na obrázku 3.3 je výsledná štruktúra paketu pre overenie paritou a na obrázku 3.4 pre overenie ostatnými algoritmami.



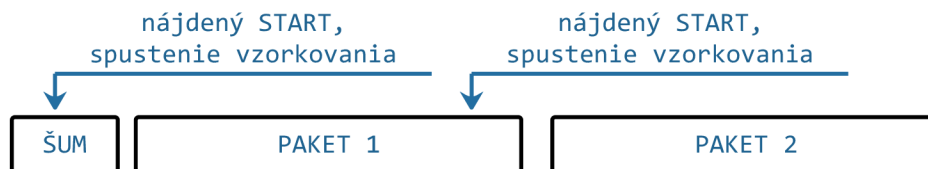
Obrázok 3.3 Schéma paketu pre paritu



Obrázok 3.4 Schéma paketu pre ostatné algoritmy

3.2 Synchronizácia

Šum, alebo dočasná strata spojenia, môže mať za následok stratu synchronizácie. Pri obyčajnom asynchrónnom UART prenose je START iba jeden bit, logická 0. Na rozdiel od UART zbernice je šum alebo pravdepodobnosť prerušenia spojenia oveľa väčšia. Preto nie je možné obyčajne „hlúpo“ vzorkovať hneď, ako sa na vstupe objaví logická 0. Napríklad môže nastať situácia, že svetelný lúč zablokuje nepriehľadnú prekážku. Ak sa prekážka odstráni v čase, kedy laser posiela paket, vzorkovanie by sa spustilo zbytočne. Takto prijaté dáta by boli nepoužiteľné. V prípade posielania viacerých paketov za sebou by došlo k strate synchronizácie a všetky nasledujúce pakety by boli poškodené. Rovnakú situáciu môže vyvolať aj šum.



Obrázok 3.5 Nesynchronizované vzorkovanie

Samozrejme, že overovací algoritmus by zahadzoval navzorkované DATA pretože by ich nedokázal overiť. Ale toto nie je dostatočné riešenie problému, keďže prenos je nefunkčný až do doby kedy prijímač začne vzorkovať na začiatku paketu. Toto je problém stratenia synchronizácie a má viacero riešení.

- Pridať toľko STOP bitov na koniec, aby ich celková dĺžka bola dlhšia, alebo v najhoršom prípade rovnako dlhá, ako je dĺžka paketu (inak povedané, zväčšiť medzeru medzi paketmi),
- signalizovať začiatok paketu START bitmi, ktoré majú rozoznateľne vyššiu frekvenciu,
- rozšíriť START bity na toľko, že je nepravdepodobný ich výskyt vo vnútri paketu.

Prvá možnosť je jednoduchá a je s ňou isté, že sa systém synchronizuje hneď po prvom nesprávne prijatom pakete. Nevýhodou je, že sa drasticky zníži rýchlosť prenosu. Priemerná dĺžka prenosu jedného paketu sa zvýši minimálne na 200 %.

Druhá možnosť je časovo lepšia. Pri zistení rýchlejšej zmeny bitov sa spustí vzorkovanie. S týmto riešením sa dá zaručiť, že prijímač nezačne vzorkovať v strede paketu a pravdepodobnosť, že ho spustí šum, je takmer nulová. Avšak vyššia frekvencia sa nedá dosiahnuť. Predpokladá sa, že systém pracuje už s maximálnou frekvenciou. To

znamená, že sa musia predĺžiť ostatné bity paketu. Aby bolo zaručene rozoznateľné, že ide o inú frekvenciu, mali by byť START bity približne 75 % veľkosti ostatných. Čiže ak sa zanedbá kratšia dĺžka STAT bitov, celková dĺžka jednej správy sa zvýši na

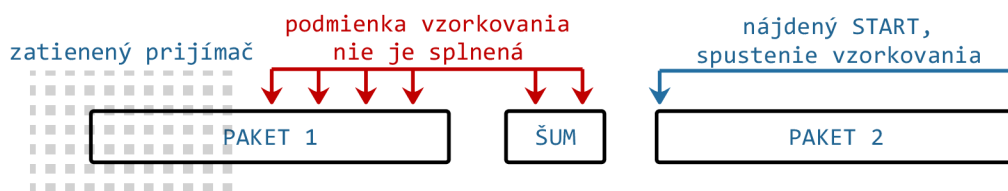
$$\frac{1}{0.75} \cdot 100 \% \approx 133 \%. \quad (3.1)$$

Teoreticky je možné ísť aj pod 133 %, záleží na tom, ako presne vie prijímač rozoznať frekvenciu meniacich sa bitov.

Posledná možnosť je určiť väčší počet START bitov, ktoré bude veľmi nepravdepodobné nájsť v ostatných častiach paketu. Avšak niekedy sa to stať môže. Výhodou je to, že dĺžka paketu sa zmení nevýrazne. Napríklad, ak by bolo START bitov 16 náhodných bitov (musia začínať aktívnou úrovňou, logickou 0), pri prenose 25 DATA bajtov s overením parity by sa paket predĺžil na

$$\frac{16 + 5 + 256 + 32}{5 + 256 + 32} \cdot 100 \% \approx 107 \%. \quad (3.2)$$

Samozrejme pri dlhšej redundancii a dátach to bude iba lepšie. Ak by pri tejto metóde nehrozilo, že vzorkovanie môže spustiť obsah paketu, oplatilo by sa ju realizovať. Avšak vzhľadom na túto skutočnosť, je lepšie použiť druhú metódu, ktorá odlišuje začiatok paketu striedavou zmenou bitov s vyššou frekvenciou na začiatku paketu. Čo sa týka dĺžky, 8 párov 01 by malo postačovať. Časový diagram prenosu je na obrázku 3.6. Ak by aj prijímač nedostal začiatok paketu, nedôjde k strate synchronizácie.



Obrázok 3.6 Časový priebeh prenosu

3.3 Komunikácia s počítačom

Mikrokontrolér bude komunikovať s počítačom cez sériový port. Rýchlosť sériového portu sa nastaví na 115200 baudov. Jeden baud je počet prenesených symbolov za jednotku času. V tomto prípade je symbol 8 bitové číslo a teda 115200 baud rate odpovedá rýchlosti 115,2 kB/s. Táto rýchlosť postačuje na prenos dát laserom, ktorý bude

v jednotkách kB/s. Neskôr sa môže zvýšiť, aby sa mohla skrátiť medzera medzi paketmi, čo by o málo pridalo na rýchlosti prenosu. Bude vhodné, aby sa program v počítači nakonfiguroval sám, podľa toho či sa do neho pripojí vysielač, alebo prijímač. Toto zabezpečí jednoduchá výmena textových reťazcov pri štarte. Mikrokontrolér hneď po inicializácii odošle textový reťazec „*Hi!-I-am-TRANSMITTER*“ v prípade vysielača, alebo „*Hi!-I-am-RECEIVER*“ v prípade prijímača. Nevadí, ak sa vysielač/prijímač pripojí ešte pred tým, ako sa zapne obslužná aplikácia. Tento reťazec sa aj tak najskôr nahrá do vyrovnávacej pamäte počítača a prečíta sa, až keď si to spustená aplikácia vyžiada. Počítač mu odpovie reťazcom ktorým mu definuje, aký protokol má použiť. V tabuľke 3.1 sú protokoly a reťazce, ktoré ich vyberajú.

Tabuľka 3.1 Textové reťazce pre jednotlivé protokoly

protokol	textový reťazec
parita	“ <i>Ok-protocol-PAR</i> ”
kontrolný súčet	“ <i>Ok-protocol-CHS</i> ”
Hammingov kód	“ <i>Ok-protocol-HAM</i> ”
cyklický redundantný súčet	“ <i>Ok-protocol-CRC</i> ”

Táto konfigurácia bude prebiehať iba na začiatku. Takže, ak bude chcieť užívateľ zmeniť protokol, musí zariadenie reštartovať.

3.4 Užívateľské rozhranie

Ako posledné treba rozhodnúť o nie menej podstatnej veci, akou je rozhranie s ktorým príde užívateľ do kontaktu. Na výber sa ponúka viacero možností. Program pre počítač je možné napísať takmer v každom programovacom jazyku, pre ktorý existujú GUI (z angličtiny „*Graphical User Interface*“) knižnice. V tomto prípade sa pôjde trochu netypickou cestou, pretože aplikácia bude vytvorená v prehliadači. Tento prístup má niektoré obmedzenia, ale zároveň aj bezkonkurenčné výhody.

Moderné prehliadače začínajú podporovať prístup k sériovému portu. Samozrejme, že rýchlosťou spracovania a odosielania dát sa nedokáže aplikácia v prehliadači vyrovnáť natívej aplikácii. V tomto prípade, kedy bude prenosová rýchlosť v jednotkách kB/s, to problém nie je. To čo sa týmto riešením získa je oveľa užitočnejšie a to je prenositeľnosť nie len medzi počítačmi, ale aj medzi operačnými systémami. Tak isto odpadá nutnosť inštalácie aplikácie do počítača, pretože webový prehliadač má v počítači nainštalovaný určite každý. Takto je jedno či sa zariadenie zapojí do Apple

počítača, Chrome Booku, Linuxu alebo Windows notebooku. Kým má počítač nainštalovaný prehliadač s podporu sériového portu, prenos bude fungovať. V čase písania tejto práce podporujú sériový port prehliadače založené na Chrómiu (Google Chrome, Microsoft Edge ...)

Celý kód bude napísaný v jednom HTML dokumente, čo trochu zneprehľadní jeho písanie. Je to kvôli tomu, že ak má byť webová aplikácia prenositeľná a spustiteľná (napríklad z USB kľúča), musia byť CSS a JS časti kódu spolu v HTML súbore. Ak by boli všetky súbory zvlášť, po otvorení HTML dokumentu by sa ich prehliadač snažil importovať, čo by sa mu nepodarilo a to kvôli ochrane údajov na disku užívateľa. Zvlášť by mohli byť jedine na serveri, ktorý by súbory poskytol.

3.5 Prenos bajtov

Vrstva na prenos bajtov bude nazvaná transportná vrstva. Táto vrstva vo vysielajúci odosiela dáta do sériového portu, ktoré dostane buď priamo, alebo z vyššej vrstvy (nasledujúca kapitola Prenos súborov). Okrem odosielaní dát do sériového portu má na strane vysielajúča ďalšie 3 úlohy. Dopĺňať nuly na koniec dát, aby všetky dáta boli násobkom 32. To ale len v prípade protokolov CRC a Hammingov kód, ktoré potrebujú 32 bitové dáta. Jej druhá úloha je časovo rozostúpiť dáta, aby sa nestalo, že vyrovnávacia pamäť sériovej komunikácie mikrokontroléru pretečie. Posledná úloha je meranie rýchlosti. Na strane prijímača okrem merania rýchlosti sa táto vrstva stará o posielanie dát zo sériového portu poslednej vrstve. Ak je posledná vrstva vypnutá, dáta posielajú do textového výstupu.

3.6 Prenos súborov

Posledná vrstva zabezpečuje prenos súborov a bude nazvaná aplikačná vrstva. Protokol, ktorým sa bude vysielajúč aj prijímač riadiť, by sa dal prirovnať k internetovému UDP protokolu. To, že prijímač nevie komunikovať späť, prináša určitý problém pri prenose dát väčších, ako dokáže naraz preniesť jeden paket. Napríklad pri prenose obrázku sa môže stať, že niektoré pakety zničí šum a neprejdú overovacím algoritmom. Počítač musí v takomto prípade vedieť, že mu chýbajú pakety, aby sa podľa toho mohol zariadiť (napríklad zahodiť dáta z už prijatých paketov). Prvoplánové riešenie je poslať z prijímača informáciu počítaču, že paket bol zlý. To má ale dva nedostatky. Prvým je, že aj keď je začiatok paketu signalizovaný zvýšenou frekvenciou a je malá

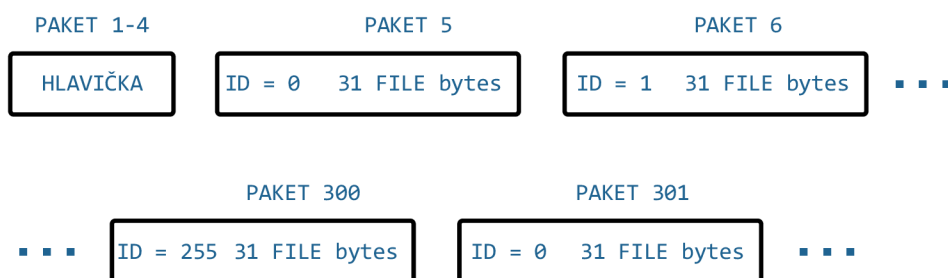
pravdepodobnosť, že ho spustí náhodný šum, nie je dobré sa na to spoliehať. Môže totiž nastať situácia, kedy sa takýto šum na vstupe objaví. Nasledovalo by vzorkovanie a prijímač by odoslal do počítača informáciu o chybné prijatom pakete aj keď by v skutočnosti išlo o šum. Ak by sa to stalo v strede prijímania súboru, počítač by ho zahodil zbytočne. Druhý nedostatok tohto riešenia je v otázke, ako má vyzerat' informácia, ktorá sa má odoslať do počítača. Mohol by to byť napríklad reťazec „RECEIVING_PACKET_FAILED“. Čo však v prípade, ak vysielač odošle dáta s rovnakou bitovou reprezentáciou? Šanca, že vysielač odošle práve tieto dáta je malá, ale znamenalo by to, že sa neprijmú. Bolo by to možné obísť, napríklad použitím dohodnutých znakov, ktoré by sa zdvojovali tak, ako používa známa C funkcia

```
int printf(const char * format,...). (3.3)
```

Existuje ale jednoduchšie riešenie, priradiť každému paketu ID. Takto bude mikrokontrolér iba „hlúpo“ posielat' to, čo príde z demodulátora a cez overenie algoritmom. Počítač pripojený k vysielaču odošle ako prvé hlavičku, v ktorej sa bude nachádzať informácia o veľkosti súboru a následne dáta v paketoch spolu s ID. V takto nastavenom, procese ak by neprešiel niektorý z paketov, počítač to odhalí a zariadi sa podľa toho. ID ale nebude číslo od 0 až po počet prenesených paketov. Jeden paket obsahuje maximálne 32 bajtov a míňať z neho viac ako jeden bit na ID je zbytočný luxus, ktorý by spomalil prenos. Preto ak bude číslo paketu N_p , ID tohto paketu sa vypočíta ako

$$ID = N_p \% 256. (3.4)$$

Cyklovať bude od 0 do 255. Šanca, že sa stratí presne 255 paketov po sebe je nereálna. Iba v takom prípade by počítač na strane prijímača neodhalil chybu. To, že na dáta ostane 31 bajtov vôbec nevadí. Prenášaný súbor sa rozdelí na 31 bajtové časti. V poslednom pakete bude pravdepodobne zvyšné miesto, ale počítač z hlavičky súboru presne vie pokiaľ má dáta brať. Na obrázku 3.7 je načrtnutý prenos súboru.



Obrázok 3.7 Protokol na prenos súborov

Pri prenášaní súboru sa bude zobrazovať progres bar, aby mal užívateľ približne predstavu, kedy sa prenos dokončí. Prijaté súbory sa uložia do dočasnej pamäte, z ktorej ich bude možné stiahnuť. Túto pamäť môže užívateľ kedykoľvek uvoľniť.

3.6.1 Štruktúra hlavičky

Pri prenose súborov je potrebné okrem samostatných dát súboru preniesť aj 3 parametre o prenášanom súbore. Tieto tri parametre sú.

- Názov súboru,
- MIME typ súboru,
- veľkosť súboru.

Tieto tri parametre bude obsahovať hlavička. Samozrejme, mohla by obsahovať aj iné parametre, ale tieto tri sú nevyhnutné. Názov súboru je v hlavičke kvôli tomu, aby sa prijatý súbor volal rovnako ako ten odoslaný. Druhý parameter je MIME typ súboru. Ten hovorí o formáte súboru. Formát súboru je možné zistiť aj z prípony, ale spoliehať sa na to nie je dobrá prax. Príponu môže meniť aj užívateľ a niektoré súbory sa pri prenose zakazujú, napríklad spustiteľné súbory počítačom. Našťastie MIME typ zistí prehliadač automaticky z dát súboru hneď potom, ako mu dá k nemu užívateľ prístup. MIME typy sú definované organizáciou Internet Engineering Task Force (komisia pre technickú stránku internetu) [29] v dokumente RFC6838 [27]. V tabuľke 3.2 je ukážka často používaných MIME typov [28]. Posledný údaj v hlavičke je veľkosť súboru v bajtoch. Veľkosť je potrebná na to aby počítač na strane prijímača vedel koľko paketov má očakávať a kedy môže vyskladať súbor z prijatých dát. Na obrázku 3.8 sú všetky 3 parametre súboru, ktorému dal užívateľ prístup.

Tabuľka 3.2 Niektoré MIME typy

prípona	druh dokumentu	MIME typ
.txt	Text, (ASCII alebo ISO 8859-n)	text/plain
.png	Portable Network Graphics	image/png
.mp4	MPEG-4	video/mp4
.pdf	Portable Document Format	application/pdf

Hlavička bude objekt posielaný v jeho textovej forme, takzvaný JSON. Ten bude mať vždy rovnakú dĺžku, 4 pakety (128 bitov). Do týchto 128 bitov sa musí zmestiť JSON vygenerovaný z objektu obsahujúceho spomínané 3 parametre. Len polovicu miesta zaberie MIME typ, ktorý by mal mať podľa špecifikácie do 64 bitov [26]. Avšak

napríklad PowerPoint prezentácia má MIME typ dlhý 73 bitov („*application/vnd.openxmlformats-officedocument.presentationml.presentation*“). Názov súboru bude obmedzený na 16 bajtov aj s príponou. Veľkosť súboru bude zaberat' do 8 bitov (10000000). Väčšie súbory nebude dovolené prenášať. Okrem tohto musí byť v hlavičke aj názov jednotlivých parametrov. Prenášaný JSON textového súboru s názvom subor.txt o veľkosti 1 kB by vyzeral „*{\"name\": \"subor.txt\", \"size\": 1024, \"type\": \"text/plain\"}*“. Ak zlyhá prenos paketov s hlavičkou, program na to príde pri prevádzaní hlavičky späť na objekt. Pre istotu sa ale skontroluje či vytvorený objekt obsahuje všetky 3 parametre.

```
> fileInput.files[0]
< File {name: "subor.txt", lastModified: 1618311957652, lastModifiedDate:
  ▼ Tue Apr 13 2021 13:05:57 GMT+0200 (Central European Summer Time), webkitRelativePath: "", size: 1024, ...} ⓘ
  lastModified: 1618311957652
  ▶ lastModifiedDate: Tue Apr 13 2021 13:05:57 GMT+0200 (Central Europea...
    name: "subor.txt"
    size: 1024
    type: "text/plain"
    webkitRelativePath: ""
  ▶ __proto__: File
```

Obrázok 3.8 Súbor analyzovaný prehliadačom

3.7 VRSTVY ZARIADENIA

Pre úplnosť návrhu je vhodné uviesť vrstvy zariadenia, tabuľka 3.3. Keďže ide o pomerne zložité zariadenie, zatriedenie do vrstiev sprehl'adni jeho výrobu. Najnižšia vrstva je samozrejme hardvérová, respektíve fyzická. Za ňou nasleduje signálová a dátová vrstva. Tieto vrstvy sú úzko späté, pretože sú obe implementované v jednom kóde, ktorý je v mikrokontroléroch. Takisto sú úzko späté aj aplikačná a transportná vrstva keďže sú obe implementované v prehliadači. Užívateľ bude mať prístup aj k transportnej vrstve okrem aplikačnej. Pri výrobe sa bude postupovať od najnižšej vrstvy po najvyššiu. Teda najskôr sa vyrobí hardvér, ku ktorému sa pridá mikrokontrolér. Po naprogramovaní oboch mikrokontrolérov sa vyrobia krabičky pre obe časti zariadenia. A ako posledné sa doprogramuje GUI.

V prehliadači bude možné posielat' dáta aj priamo transportnou vrstvou. Bud' cez textové pole, alebo zo súboru. Prístup k tejto vrstve je hlavne kvôli testovaniu a posielaniu UTF-8 textu. Používaním tejto vrstvy sa môže napríklad stať, že po odoslaní desiatich 32

bajtových UTF-8 reťazcov, dorazí iba 9 a transportná vrstva to neodhalí. Preto je nad ňou aplikačná vrstva. V tejto vrstve sa budú dáta prenášať len súbory. Táto vrstva si bude sama kontrolovať prenos. Takto zabezpečí, že prijaté súbory nebudú poškodené spôsobom vysvetleným v predchádzajúcej sekcii Prenos súborov.

Tabuľka 3.3 Vrstvy zariadenia

vrstva	funkcia vysielča	funkcia prijímača	miesto implementácie
aplikačná	odosielanie súborov	prijímanie súborov	webový prehliadač (JavaScript/ES6)
transportná	odosielanie dátových reťazcov	prijímanie dátových reťazcov	
dátová	prijímanie dát z PC, generovanie redundancie	overovanie dát algoritmami, odosielanie dát do PC	mikrokontrolér (C/C++)
signálová	delenie dát na bity, odosielanie dát do modulátora	hľadanie začiatku paketu, vzorkovanie paketu, spájanie bitov	
fyzická	laserová dióda, modulátor	solárny panel, demodulátor	DPS

4 REALIZÁCIA HARDVÉROVEJ ČASTI

V tejto kapitole je predstavený praktický návrh elektroniky vysielača a prijímača. Všetky hodnoty súčiastok a spôsoby riešenia sú podopreté výpočtom. Okrem toho je táto kapitola doplnená o merania na následne skonštruovanej elektronike. Na konci bude navrhnutá krabica pre vysielač a prijímač v 3D návrhovom prostredí.

4.1 Laserová dióda

Laserová dióda, musí spĺňať niekoľko parametrov. Ako prvý a najdôležitejší je bezpečnosť. Laser vie byť nebezpečný pre oči, hlavne ak je jeho vlnová dĺžka z neviditeľného spektra. Rohovkový reflex v takomto prípade oko neochráni [17]. Na obrázku 1.2 je senzitivita kremíkovej diódy. So zvyšujúcou sa vlnovou dĺžkou má tendenciou stúpať, takže najvhodnejšia vlnová dĺžka bude v oblasti červenej farby. Tam je laser ešte viditeľný a dióda (v tomto prípade solárny panel) má stále dobrú senzitivitu. Zároveň by mal byť laser z viditeľného spektra, aby bolo možné zamieriť na prijímač. Optický výkon bol zmeraný v laboratóriu optoelektroniky. Pri prúde 25 mA má hodnotu 5 mW. Na laseri je pri tomto prúde 2,25 V čiže jeho efektívnosť je

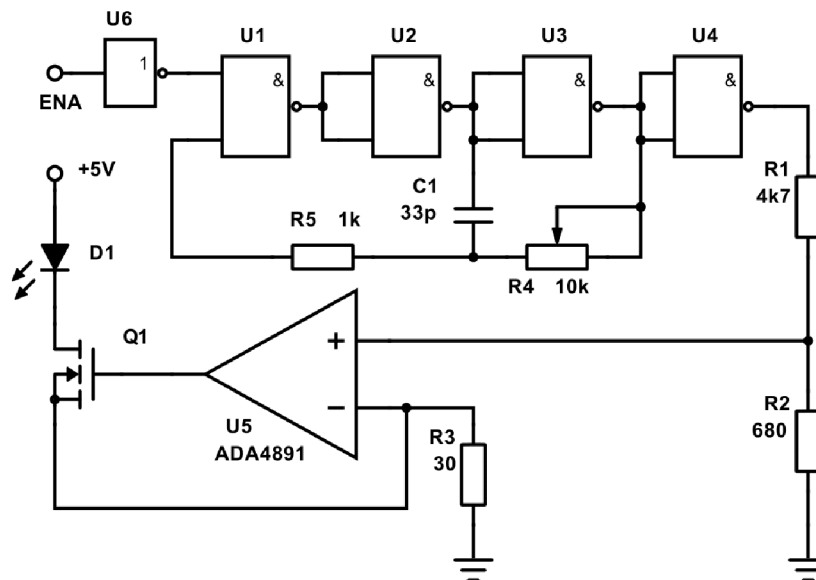
$$\eta = \frac{P_{opt}}{P_{tot}} \cdot 100 \% = \frac{5 \cdot 10^{-3}}{25 \cdot 10^{-3} \cdot 2,25} \cdot 100 \% \approx 8,9 \%. \quad (4.1)$$

Aby laser spĺňal triedu R3 (viditeľný laser do 5 mW) a bol považovaný za bezpečný pri opatrnom zaobchádzaní [18], pracovný prúd bude znížený na 20 mA. Menší prúd zároveň predĺži životnosť laseru. Teoreticky by nebolo potrebné napájať diódu z prúdového zdroja a použiť na obmedzenie prúdu iba rezistor, pretože dióda nebude pracovať plne v kontinuálnom režime. Avšak pre ochranu a rýchlosť zapnutia (1.2.1) je to žiadúce.

4.2 Elektronika vysielača

Digitálny signál sa moduluje metódou ASK. Je teda potrebné vytvoriť modulátor, ktorý je možné spustiť a zastaviť bez dobehu. Jeho signál bude zapínať a vypínať prúdový zdroj pre laser. Navrhnutých bolo viacero riešení, harmonické aj digitálne. Harmonické majú výhodu v tom, že na výstupe je iba jedna harmonická zložka. Zatiaľ čo digitálne vytvárajú teoreticky nekonečne veľa harmonických na hranách prechodu signálu. Na druhej strane, digitálne oscilátory potrebujú menší počet súčiastok, majú presne definované napäťové úrovne a všeobecne sa na nich ľahšie doladuje kmitočet. Pre tieto dôvody bude vysielač

zhotovený z digitálneho oscilátora. Použitý bude kruhový oscilátor z NAND členov zapojených ako invertory, konkrétne obvod SN74HC00. V jednoduchosti vysvetlené, ide o kruhové zapojenie nepárneho počtu NAND členov za sebou. Keďže takáto konfigurácia nemá stabilný stav, logická 0 a 1 rotujú dookola [13]. Frekvencia a päťdesiatpercentná striedavo nabíja a vybíja. Dobré je poznamenať, že aj keď sa na kondenzátore teoreticky vytvorí 2-krát väčšie napätie (prakticky nie je 2-krát väčšie, pretože logické členy sa preklapia skôr ako pri plnom napájacom napätí) ako je napájacie napätie, logickému obvodu to nevadí, pretože na vstupe má ochranné diódy. Tieto diódy (z angličtiny „clamping diodes“) orežú vyššie napätie ako je napájacie, takže nedôjde k zničeniu CMOS štruktúry logického člena. Na obrázku 4.1 je výsledný oscilátor aj so zdrojom prúdu.



Obrázok 4.1 Modulátor

Obvod je napájaný napätím 5 V z USB zbernice. Členy NAND $U1$ až $U4$ tvoria kruhový oscilátor. Trimrom $R4$ sa nastavuje frekvencia. Hodnota τ RC článku je asi 63 % maximálnej hodnoty. Obvod SN74HC00 sa preklápa približne pri 3,5 V (hodnota V_{IH} z technického listu [20]) pri napájacom napätí 5 V. Čo je v percentách

$$U_{R2} = \frac{100 \%}{V_{CC}} \cdot U_{IH} = \frac{100 \%}{5} \cdot 3,5 = 70 \%. \quad (4.2)$$

Takže hodnota τ takmer odpovedá úrovni preklapenia logického obvodu. Za jednu periódu sa kondenzátor raz nabije a vybije. Čiže ak $\tau \approx U_{R2}$, minimálna frekvencia obvodu s 10 k Ω trimrom bude

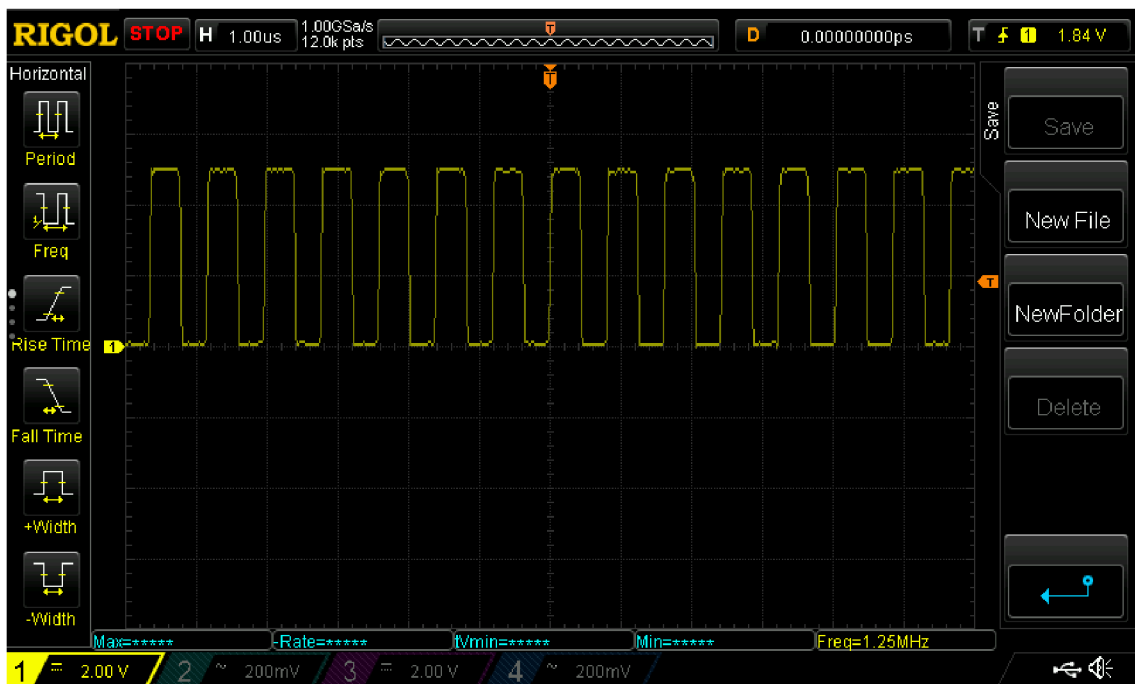
$$F_{min} = \frac{1}{2 \cdot R_4 \cdot C_1} = \frac{1}{2 \cdot 10 \cdot 10^3 \cdot 33 \cdot 10^{-12}} \approx 1.51 \text{ MHz.} \quad (4.3)$$

To by malo byť dostatočným minimom. Samozrejme výpočet je len približný, ale pre odhad postačujúci. Presná hodnota sa aj tak dostaví podľa rezonančnej frekvencie prijímača. Odmeraná minimálna frekvencia je na obrázku 4.2. Rezistor R_5 obmedzuje prúd do ochrannej diódy. Delič R_1 a R_2 delí napätie z 5 V na 630 mV

$$U_{R2} = \frac{R_2}{R_1 + R_2} \cdot V_{CC} = \frac{680}{4700 + 680} \cdot 5 \approx 630 \text{ mV.} \quad (4.4)$$

Operačný zosilňovač udržiava toto napätie na odpore R_3 a teda zabezpečuje konštantný prúd laserom, ktorý je

$$I_{R3} = \frac{U_{R2}}{R_3} = \frac{0,630}{30} \approx 21 \text{ mA.} \quad (4.5)$$



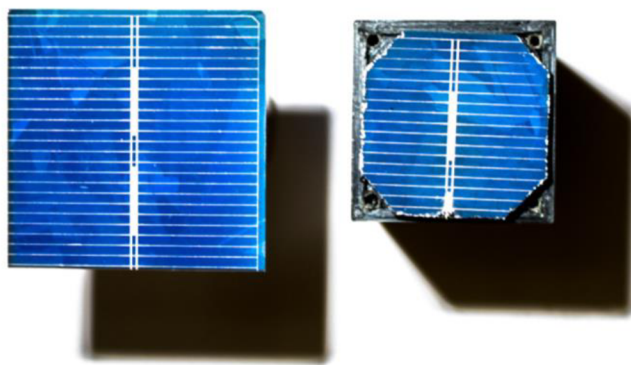
Obrázok 4.2 Minimálna nastaviteľná frekvencia kruhového oscilátoru

Oscilácie sa vypínajú logickou nulou na prvom vstupe NAND člena $U1$. V dizajne je aktívna úroveň 0, takže pred tento vstup musí ísť invertor. Vstup ENA je ovládaný mikrokontrolérom. Všetky integrované obvody majú pri sebe blokovací kondenzátor, ktorý v schéme nie je zahrnutý. Kompletná schéma modulátora aj s návrhom dosky plošného spoja je v prílohe, vyexportovaná z návrhového prostredia EAGLE.

4.3 Solárny panel

Namiesto premeny slnečnej energie na elektrickú, bude v tomto prípade solárny panel použitý na prijímanie signálu z laseru a to kvôli jeho veľkej ploche. Solárne panely sú veľmi lacné, v čom tkvie ich výhoda. Konkrétne to bolo 0,07 € v prepočte na jeden kus. Ide o polykryštalický solárny panel, takže elektróny majú približne o rád menšiu pohyblivosť, ako má monokryštalický. Bolo by možné použiť aj fotodiódu s lepšími frekvenčnými vlastnosťami, ale na to je potrebná optika, ktorá by zariadenie predražila. Pôvodný solárny panel má rozmery 39 x 39 mm. Bol však orezaný na 28 x 28 mm, okrem toho boli orezané aj rohy panelu. Je to z dôvodu zníženia parazitnej kapacity a hlavne, aby sa rozmerovo vošiel do krabičky.

Ak ide o rezanie kremíku, väčšinou sa používa diamant s rôznymi formami použitia. Môže byť na hrote nástroja, nanosený na reznom kotúči, poprípade aj na kovovom vlákne. V tomto prípade sa ale rezal kotúčom na železo, pretože pokusy rezania diamantovým hrotom boli neúspešné. Zrejme išlo však viac o brúsenie ako o rezanie. Uhlová brúska bola pevne prichytená na stole a jemným pohybom solárneho panelu proti kotúču bolo z panelu odoberané až po vyznačenú čiaru. Na obrázku 4.3 je panel pred a po rezaní. Tento spôsob bol relatívne rýchly, presný a zároveň jedený, s ktorým sa podarilo orezať panel bez jeho prasknutia.



Obrázok 4.3 Solárny panel pred a po rezaní

Výkon laseru je už zmienených 5 mW pri 25 mA. Pri svietení laserom na panel bol skratový prúd panelu 2 mA. Senzitivita panelu je

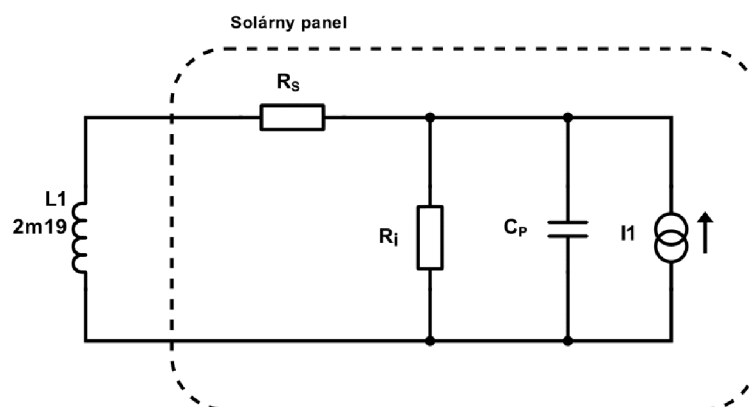
$$R_{\lambda} = \frac{I}{P} = \frac{2 \cdot 10^{-3}}{5 \cdot 10^{-3}} = 0,4 \text{ AW}^{-1}. \quad (4.6)$$

To aj približne odpovedá grafu 1.2, pretože vlnová dĺžka laseru je 650 nm. Vonkajšia kvantová efektivita je

$$EQE = R_{\lambda} \frac{h \cdot c}{\lambda \cdot q} = \frac{6.626 \cdot 10^{-34} \cdot 3 \cdot 10^8}{650 \cdot 10^{-9} \cdot 1.602 \cdot 10^{-19}} \approx 0,76 [-]. \quad (4.7)$$

Teda 76 % fotónov, ktoré vychádzajú z laseru, vyrazí pár elektrón-diera a prispeje k celkovému prúdu. To ale neznamená, že efektivita solárneho panelu je 76 %, pretože kvantová efektivita sa počíta z prúdu a nie z výstupného výkonu. Elektróny neudržia prebytočnú časť svojej energie a minú ich vo forme tepla pri zrážkach s kryštálovou mriežkou. Takisto, kvantová efektivita sa počíta iba z prúdu dier alebo elektrónov. Pretože keď fotón vyrazí pár elektrón-diera, uvoľní iba jeden náboj. Diera je len miesto ktoré zanechá (aj napriek tomu, že sa toto miesto posúva).

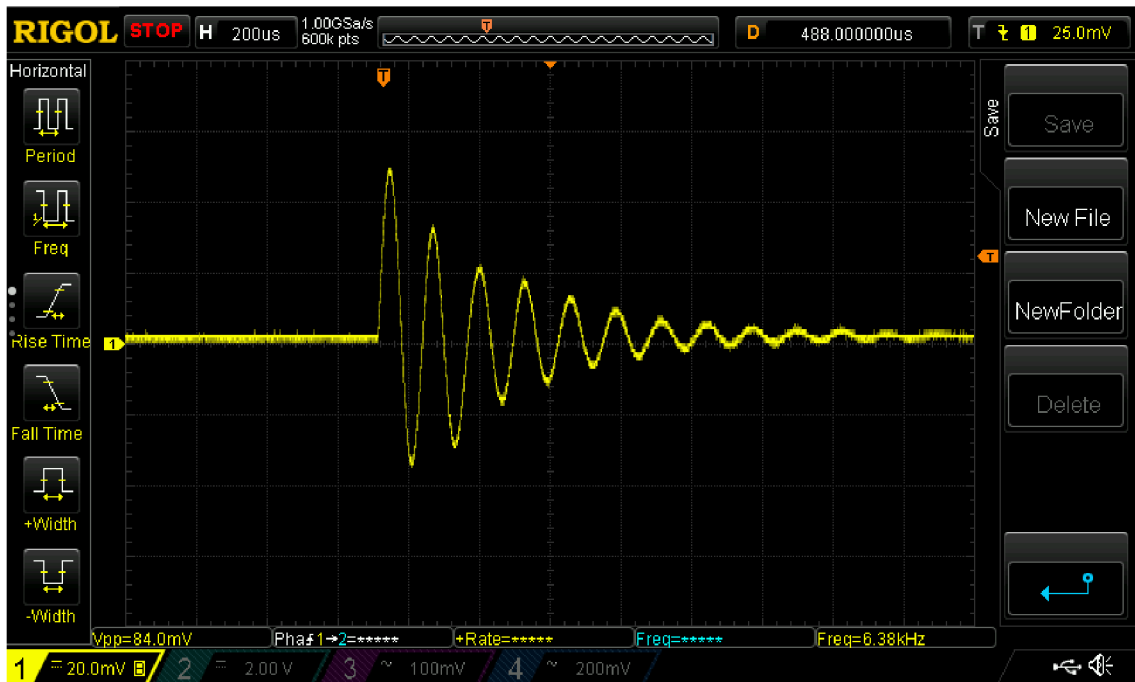
Náhradnú schému solárneho panelu je vidieť na obrázku 4.4. Ako C_p je označená parazitná kapacita, R_s je sériový odpor, R_i je vnútorný odpor a I_l je prúd vyvolaný osvetlením. Vnútorný odpor solárneho panelu nie je konštantný, ale mení sa od pretekajúceho prúdu. Pri nulovom prúde je maximálny (rádovo v M Ω) a so zvyšujúcim sa prúdom klesá.



Obrázok 4.4 Náhradná schéma solárneho panelu

K solárnemu panelu je pripojená cievka. Tá spolu s parazitnou kapacitou tvorí paralelný rezonančný obvod, ktorý je prvým stupňom prijímača. Neskôršie bude cievka vymenená za menšiu, pre vyšší rezonančný kmitočet. Teraz je potrebné odmerať parazitnú kapacitu. Na to postačí cievka o veľkosti 2.19 mH. Náhlým zhasnutím svetla

nad solárnym panelom sa rezonančný obvod rozkmitá a zo známej frekvencie a indukčnosti sa vypočíta parazitná kapacita.



Obrázok 4.5 Kmitanie pridanej cievky a parazitnej kapacity pri nulovom predpätí

Ak je frekvencia kmitania 6,38 kHz, potom parazitná kapacita tohto solárneho panelu je

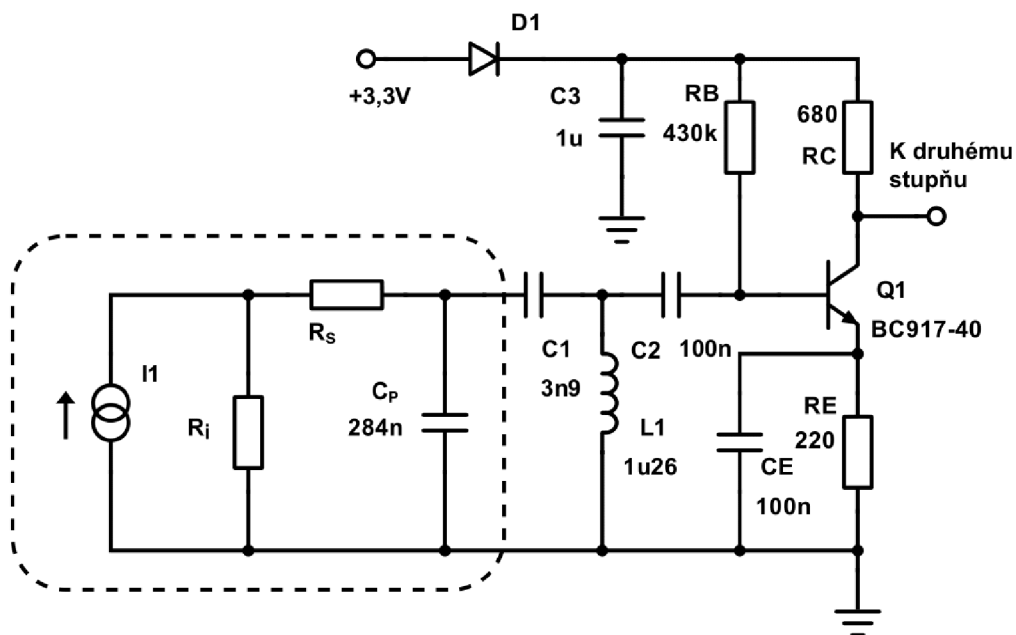
$$C_P = \frac{1}{(2\pi f)^2 \cdot L} = \frac{1}{(2\pi \cdot 6,38 \cdot 10^3)^2 \cdot 2,19 \cdot 10^{-3}} = 284 \text{ nF}. \quad (4.8)$$

Problém je, že sa kapacita mení od napätia a pretekajúceho prúdu podľa vzorcov (1.4) a (1.5). Túto nechcenú zmenu kapacity je možné ošetriť dvoma spôsobmi. Prvý spôsob je pridanie externého kondenzátora do série so solárnym panelom. Ak bude tento kondenzátor dostatočne malý, môže sa parazitná kapacita zanedbať. Zároveň sa zväčší rezonančný kmitočet a s ním maximálna prenosová rýchlosť. Je to ale za cenu svetelnej saturácie. Ak bude na panel totiž svietiť intenzívne statické osvetlenie, tým, že je oddelený externým kondenzátorom, nemá kadiaľ tento prúd pretekať. Môže pretekať len vnútram solárneho panelu. Tým sa znižuje spomínaný vnútorný odpor až do bodu, kedy bude nezanedbateľný oproti vstupnej impedancii rezonančného obvodu. Takže prúd nebude tiecť do rezonančného obvodu a ten prestane oscilovať. Druhou možnosťou je vytvoriť rezonančný obvod iba z parazitnej kapacity C_P . Potom privedením záporného predpätia na panel by sa zmena kapacity stabilizovala. Zároveň by sa tým predišlo svetelnej saturácii, pretože prúd vytvorený statickým osvetlením by bol skratovaný

zdrojom záporného predpätia. V tomto prípade bude ale lepšie zrealizovať prvú možnosť a to z dôvodu, že väčšia rýchlosť znamená širšiu použiteľnosť zariadenia. A zároveň by svetlo dopadajúce na panel muselo byť veľmi intenzívne, aby ho vyradilo z prevádzky (kvôli vysokému rozdielu medzi malou impedanciou rezonančného obvodu a vysokým odporom paralelného odporu solárneho panelu). Faktor kvality rezonančného obvodu nesmie byť vysoký, aby obvod nemal veľkú zotrvačnosť. Inak by boli veľké časové rozdiely v bitoch logických núl a jednotiek. Navyše svetelnej saturácií je možné predísť aj krytkou v tvare krátkeho tunelu pred prijímačom. Kým ale nepôjde o slnečné žiarenie, nemal by to byť problém.

4.4 Elektronika prijímača

Solárny panel bude teda oddelený kondenzátorom pre zvýšenie kmitočtu. Schéma prvého stupňa demodulátora je na obrázku 4.6. Signál z rezonančného obvodu zosilňuje tranzistorový zosilňovač v konfigurácii SE. Použitý mikrokontrolér má 3,3 V logiku. Elektronika demodulátora bude napájaná z 3,3 V LDO regulátora (z angličtiny „*Low-Dropout*”). Aj ak by bol mikrokontrolér 5 V, je regulátor potrebný na odstránenie šumu z napájania. Bez tejto aktívnej filtrácie by tranzistor zosilňoval aj šum a obvod by nefungoval. Napájanie tranzistorového zosilňovača je ešte oddelené rýchlou diódou (z angličtiny „*fast recovery diode*”), ktorá nabíja 1 μF kondenzátor, ako jeho lokálny zdroj energie, pre ešte lepšiu filtráciu.



Obrázok 4.6 Prvý stupeň zosilňovača

Rezonančný obvod bude mať v stave rezonancie veľmi malú impedanciu, takže takmer celý prúd potečie von z panelu a takmer žiadny paralelným odporom. Avšak ako už bolo povedané, externým osvetlením sa môže znížiť hodnota paralelného odporu natoľko, že prúd bude tiecť ním a obvod prestane fungovať. Táto situácia nastane až pri veľkom osvetlení, keďže sériový odpor je rádovo v Ω a paralelný rádovo v $M\Omega$. Kapacita C_p je oveľa väčšia ako C_I , preto s ňou nie je potrebné uvažovať vo výpočte rezonančnej frekvencie. Takisto tranzistorový zosilňovač zaťažuje svojou parazitnou kapacitou rezonančný obvod a jemne mení rezonančný kmitočet. Rezonančná frekvencia je približne

$$f_{rez} = \frac{1}{2\pi\sqrt{LC}} = \frac{1}{2\pi\sqrt{1,26 \cdot 10^{-6} \cdot 3,9 \cdot 10^{-9}}} \approx 2,27 \text{ MHz}. \quad (4.9)$$

Prúd tranzistorom je zvolený na 1,5 mA. Použitý tranzistor je BC817-40. Tento tranzistor má vysokú β , takže prúd $I_C \approx I_E$. Na odpore R_E sa nechá približne 1/8 napájacieho napätia (12,5 %). Začne sa s výpočtom R_E a r'_E

$$R_E = \frac{0,125 \cdot (V_{CC} - U_D)}{I_C},$$

$$R_E = \frac{0,125 \cdot (3,3 - 0,6)}{1,5 \cdot 10^{-3}} = 225 \Omega \rightarrow R_E = 220 \Omega, \quad (4.10)$$

$$r'_E = \frac{V_T}{I_C} = \frac{26 \cdot 10^{-3}}{1,5 \cdot 10^{-3}} = 17,3 \Omega.$$

Na R_E bude napätie

$$U_{RE} = I_C \cdot R_E = 1,5 \cdot 10^{-3} \cdot 220 = 0,33 \text{ V}. \quad (4.11)$$

Je najlepšie, ak je pracovný bod v strede napájacieho napätia posunutý o napätie na R_E , teda na odpore R_C bude napätie

$$U_{RC} = \frac{V_{CC} - U_D - U_{RE}}{2} = \frac{3,3 - 0,6 - 0,33}{2} = 1,185 \text{ V}. \quad (4.12)$$

Ak ním má pretekať prúd 1,5 mA, musí mať hodnotu

$$R_C = \frac{U_{RC}}{I_C} = \frac{1,185}{1,5 \cdot 10^{-3}} = 790 \Omega \rightarrow R_C = 680 \Omega. \quad (4.13)$$

Parameter β je z technického listu v rozsahu od 250 do 600, po zmeraní multimetrom je β okolo 400 (teplota okolia bola 25 °C). R_B teda musí byť

$$R_B = \frac{V_{CC} - U_D - U_{BE} - U_{RE}}{I_C / \beta}, \quad (4.14)$$

$$R_B = \frac{3,3 - 0,6 - 0,65 - 0,33}{1,5 \cdot 10^{-3} / 400} \approx 459 \text{ k}\Omega \rightarrow R_C = 430 \text{ k}\Omega.$$

Kapacita medzi bázou a kolektorom tranzistoru BC817 je 10 pF. Zosilnenie zapojenia je

$$A_{U1} = \frac{R_E}{r'_E} = \frac{680}{17,3} \approx 40 [-], \quad (4.15)$$

teda kapacita medzi bázou a emitorom sa javí 40-krát väčšia kvôli Millerovmu javu (1.4.1). Bude nazvaná C_M a má hodnotu 400 pF. Táto kapacita uberá zo signálu na vstupe a je hlavným dôvodom poklesu vstupnej impedancie pri vyšších frekvenciách. Signál ešte prechádza odporom R_C , ktorý je sériovo za C_M . Číselne je táto impedancia pri 2,27 MHz

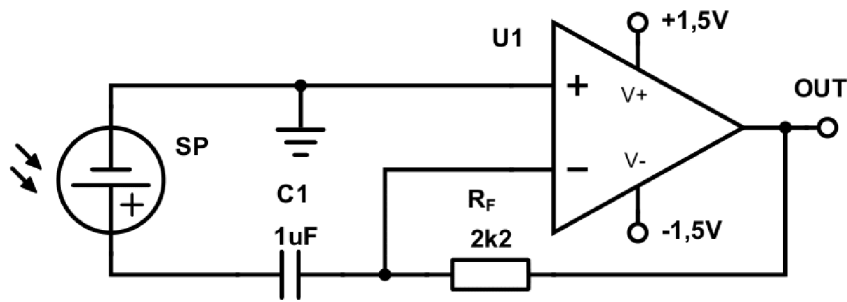
$$Z_{IN} = \frac{1}{\frac{1}{\frac{1}{2\pi f \cdot C_M} + R_C} + \frac{1}{R_B} + \frac{1}{\beta \cdot r'_e}}, \quad (4.16)$$

$$Z_{IN} = \frac{1}{\frac{1}{\frac{1}{2\pi \cdot 2,27 \cdot 10^6 \cdot 400 \cdot 10^{-12}} + 680} + \frac{1}{430 \cdot 10^3} + \frac{1}{400 \cdot 17,3}}}$$

$$Z_{IN} \approx 760 \Omega.$$

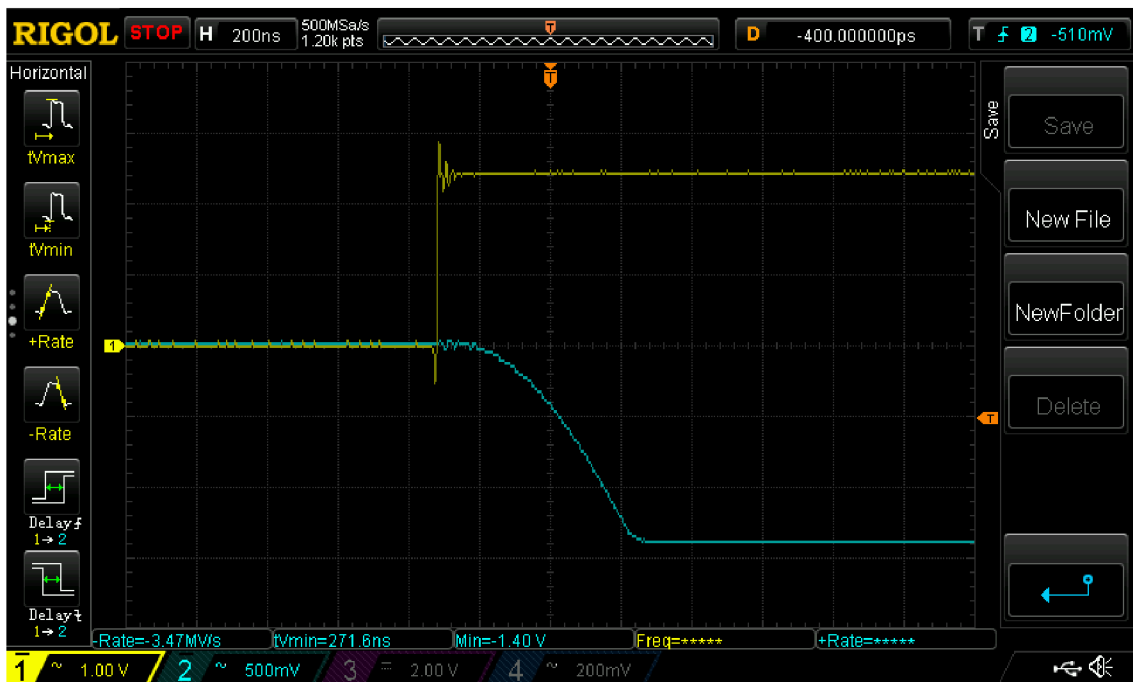
Tak isto táto kapacita jemne mení rezonančnú frekvenciu, problém to ale nie je, lebo frekvencia vysielača sa bude dolad'ovať, aby prijímač rezonoval čo s najväčšou amplitúdou.

Minimálna prenosová rýchlosť dát má byť 1 kB/s. Limitujúci faktor je intenzita prijatého signálu, respektíve jeho pomer k šumu. Aj keď je na vstupe rezonančný obvod, ktorý selektívne vyberie chcenú frekvenciu, nie je možné donekonečna zosilňovať signál stratený v šume. Preto je vhodné výpočtom zistiť aké bude pri frekvencií 2,27 MHz napätie na výstupe. Prvý krok je odmerať rýchlosť nábehu prúdu na solárnom článku, na čo postačí operačný zosilňovač zapojený ako derivačný člen.



Obrázok 4.7 Derivačný člen, zapojenie na zmeranie rýchlosti nábehu prúdu

Zámerom tohto zapojenia je skonvertovať prúd vychádzajúci z článku na merateľné napätie. Potom zo známeho odporu R_F a napätia na výstupe sa vypočíta rýchlosť nábehu prúdu. Dôležité na tomto zapojení je, že na solárnom paneli je vždy 0 V a teda celý prúd tečie von cez rezistor R_F a nie vnútorným odporom článku. Na meranie je potrebný symetrický zdroj.



Obrázok 4.8 Prúd solárneho panelu vyvolaný jednotkovým skokom laserového lúča

Žltý je priebeh napätia na laserovej dióde, tá je napájaná zdrojom prúdu pre čo najkratší čas zapnutia. Modrý priebeh je napätie na výstupe operačného zosilňovača. Je dôležité, aby mal operačný zosilňovač dostatočne veľkú rýchlosť prebehu. Operačný zosilňovač saturoval na zápornej vetve napájacieho napätia, ale z predchádzajúceho úseku je možné zmerať rýchlosť nábehu prúdu. Keďže ide o invertujúce zapojenie, jednotkový skok laseru na článku sa prejaví poklesom napätia na výstupe, ale znamienko v tomto prípade nie je podstatné. Zmena napätia bola 3,47 MV/s, z čoho sa rýchlosť prúdu vypočíta

$$SR_i = \frac{SR_u}{R_f} = \frac{3,47 \cdot 10^6}{2200} \approx 1577 \text{ As}^{-1}. \quad (4.17)$$

Každý harmonický signál bez fázového posunu je možné zapísať v tvare

$$i(t) = I_{max} \cdot \sin(2\pi f \cdot t). \quad (4.18)$$

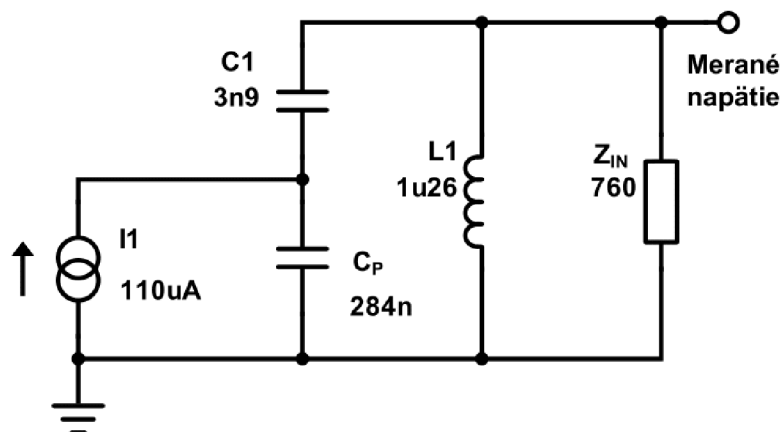
Deriváciou tohto výrazu podľa času sa získa vzťah medzi strmosťou, amplitúdou a frekvenciou. Strmosť je zmeraná 1577 As^{-1} , preto je možné dopočítať maximálnu amplitúdu pre danú frekvenciu

$$i'(t) = \partial \frac{I_{max} \cdot \sin(2\pi f \cdot t)}{\partial t} = I_{max} \cdot 2\pi f \cdot \sin(2\pi f \cdot t). \quad (4.19)$$

Pre nosnú frekvenciu 2,27 MHz je amplitúda prúdového zdroja solárneho panelu

$$\begin{aligned} i'(0) &= SR_i, \\ I_{max} \cdot 2\pi f \cdot \sin(2\pi f \cdot t) &= 1577, \\ I_{max} &= \frac{1577}{2\pi \cdot 2,27 \cdot 10^6 \cdot \sin(2\pi f \cdot 0)}, \\ I_{max} &\approx 110 \mu\text{A}. \end{aligned} \quad (4.20)$$

Vplyv odporov solárneho panelu, ako bolo povedané, sa zanedbáva. Tak isto pred tranzistorovým zosilňovačom je oddeľovací kondenzátor C2, ten rovnako nemá vplyv na výstupné napätie pri vysokých frekvenciách (nad 1 MHz), takže sa zanedbáva. Rezonančný obvod sa teda zjednoduší na schému na obrázku 4.9. Zaťaženie, ktoré vyvolá tranzistorový zosilňovač je označené ako Z_{IN} .



Obrázok 4.9 Zjednodušená schéma rezonančného obvodu so záťažou

Aj keď je toto zaťaženie nie úplne odporové, bude sa s ním počítať ako s odporovým, jemne sa kvôli tomu bude líšiť faktor kvality od reálu a aj rezonančná frekvencia. Faktor kvality je rezonančného obvodu je

$$Q = R \sqrt{\frac{C_1 || C_P}{L}} = 760 \sqrt{\frac{3,9 \cdot 10^{-9} \cdot 284 \cdot 10^{-9}}{3,9 \cdot 10^{-9} + 284 \cdot 10^{-9}}} \approx 42 [-], \quad (4.21)$$

čo je relatívne dobrá hodnota. Rezonančný obvod nebude dlho kmitať po tom, ako sa prestane budiť. A teraz výpočet spomínaného napätia. Ide o delič prúdu medzi kondenzátorom C_P a druhou vetvou. Napätie sa vypočíta ako prechádzajúci prúd vynásobený impedanciou paralelného zapojenia L a Z_{IN} . V tomto výpočte sa musí počítať v komplexnej rovine s fázormy a amplitúdu sa získa absolútnou hodnotou. Je to teda

$$U_{REZ-AC-OUT} = \left| I_1 \cdot \frac{Z_{CP}}{Z_{CP} + Z_{C1} + (Z_{L1} || Z_{IN})} \cdot Z_{L1} || Z_{IN} \right|, \quad (4.22)$$

$$\left| I_1 \cdot \frac{\frac{1}{2\pi f \cdot C_P \cdot j}}{\frac{1}{2\pi f \cdot C_P \cdot j} + \frac{1}{2\pi f \cdot C_1 \cdot j} + \left(\frac{2\pi f \cdot L_1 \cdot j \cdot Z_{IN}}{2\pi f \cdot L_1 \cdot j + Z_{IN}} \right)} \cdot \frac{2\pi f \cdot L_1 \cdot j \cdot Z_{IN}}{2\pi f \cdot L_1 \cdot j + Z_{IN}} \right|.$$

Pre komplexnosť a neprehľadnosť už nebudú dosadené konkrétne hodnoty ². S frekvenciou 2,27 MHz vychádza amplitúda na výstupe rezonančného obvodu

$$U_{REZ-AC-OUT} \approx 970 \mu V. \quad (4.23)$$

Po zosilnení bude napätie na kolektore tranzistora

$$U_{Q1-AC-OUT} = U_{REZ-AC-OUT} \cdot A_{U1} = 970 \cdot 10^{-6} \cdot 40 = 38,8 mV. \quad (4.24)$$

Túto hodnotu je potrebné porovnať so šumom a overiť, či je oproti nemu dostatočne veľká. Integrálna hodnota šumového napätia na výstupe bipolárneho tranzistora sa vypočíta z dvoch šumových príspevkov. Z prúdového šumu kolektoru a šumu bázoového odporu. Tretí, prúdový šum báze, sa bude zanedbávať. Prúdový šum kolektoru je

² [Odkaz na výpočet](#)

$$i_{nc} = \sqrt{2 \cdot q \cdot I_C} = \sqrt{2 \cdot 1,602 \cdot 10^{-19} \cdot 1,5 \cdot 10^{-3}} \approx 21,9 \text{ pA}/\sqrt{\text{Hz}}. \quad (4.25)$$

Tranzistor BC817-40 má bázový odpor r_{bb} približne 30 [21]. Šum bázového odporu je

$$u_{nrbb} = \sqrt{4kT \cdot r_b} = \sqrt{4 \cdot 1,38 \cdot 10^{-23} \cdot 300 \cdot 30} \approx 0,705 \text{ nV}/\sqrt{\text{Hz}}. \quad (4.26)$$

Aby sa obidva šумы dali spojiť do jedného, prepočíta sa šum bázového odporu na prúd kolektorom cez transkonduktanciu

$$gm = \frac{1}{r'_e} = \frac{1}{17,3} \approx 57,8 \text{ mS}, \quad (4.27)$$

$$i_{nrbb} = gm \cdot u_{nrbb} = 57,8 \cdot 10^{-3} \cdot 0,705 \cdot 10^{-9} \approx 40,8 \text{ pA}/\sqrt{\text{Hz}}.$$

Šum je náhodný jav, preto sa sčítajú nekorelované

$$i_{ntot} = \sqrt{i_{nrbb}^2 + i_{nc}^2} = \sqrt{(40,8 \cdot 10^{-12})^2 + (21,9 \cdot 10^{-12})^2}, \quad (4.28)$$

$$i_{ntot} \approx 46,3 \text{ pA}/\sqrt{\text{Hz}}.$$

Tento prúd prechádza odporom R_C , takže napätie sa získa ich vynásobením

$$u_{ntot} = R_C \cdot i_{ntot} = 680 \cdot 46,3 \cdot 10^{-12} \approx 31,5 \text{ nV}/\sqrt{\text{Hz}}. \quad (4.29)$$

Lomový kmitočet je pre NPN tranzistory veľmi nízky (typicky v jednotkách Hz), preto bude použitá hodnota $f_k = 1 \text{ Hz}$. Integrálna hodnota v rozsahu 10 Hz až 10 MHz je

$$U_N = u_{ntot} \cdot \sqrt{f_2 + f_k \cdot \ln\left(\frac{f_2}{f_1}\right)}, \quad (4.30)$$

$$U_N = 31,5 \cdot 10^{-9} \cdot \sqrt{10 \cdot 10^6 + \ln\left(\frac{10 \cdot 10^6}{10}\right)} \approx 99,6 \text{ } \mu\text{V}.$$

Napätie z rezonančného obvodu na výstupe prvého stupňa je teda približne 390-krát väčšie ako šum. Ak by bol šum porovnateľný s chceným signálom, bolo by náročné s takýmto signálom pracovať a muselo by sa buď zvýšiť zosilnenie, znížiť frekvencia, alebo použiť nízkošumový tranzistor (prípadne sa vysporiadať s nízkou Z_{IN} impedanciou, ktorá uberá zo vstupného signálu, napríklad iným zapojením, alebo takisto iným tranzistorom). Zosilnenie sa odvíja od napájacieho napätia a ak by bolo potrebné ísť nad 5 V, možnosť by bola použiť DC-DC menič. Tým by sa obvod zbytočne skomplikoval. Jednoduchšie by bolo znížiť frekvenciu za cenu rýchlosti prenosu.

Je jasné, že napätie 33,2 mV nie je dostatočné pre digitálne spracovanie. Takže za prvý tranzistorový stupeň pôjde druhý, s rovnakým pracovným bodom. Schéma je na obrázku 4.11. Vstupný odpor druhého stupňa je rovnaký ako prvého stupňa (760 Ω), zatiaľ čo výstupný odpor z prvého stupňa je približne R_C . Toto tvorí delič napätia, takže napätie, ktoré sa bude zosilňovať je

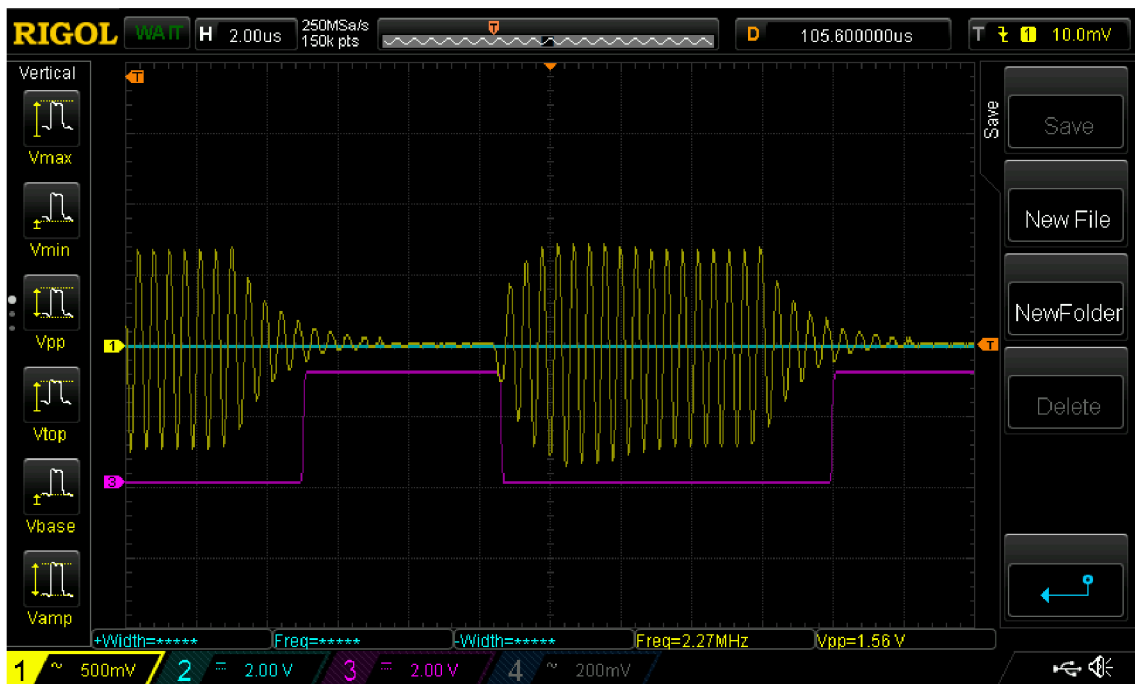
$$U_{Q2-AC-IN} = U_{Q1-AC-OUT} \cdot \frac{Z_{IN}}{Z_{IN} + R_C} = 38,8 \cdot 10^{-3} \cdot \frac{760}{760 + 680}, \quad (4.31)$$

$$U_{Q2-AC-IN} \approx 20,47 \text{ mV}.$$

Zosilnenie je rovnaké, takže napätie na výstupe druhého stupňa je

$$U_{Q2-AC-OUT} = A_{U2} \cdot U_{Q2-AC-IN} = 40 \cdot 20,47 \cdot 10^{-3} \approx 0,82 \text{ V}. \quad (4.32)$$

Je potrebné pamätať na to, že ide o amplitúdu, čiže V_{pp} je 2-krát toľko (1,64 V). Odmeraná hodnota je na obrázku 4.10.



Obrázok 4.10 Zosilnený signál dvojstupňovým tranzistorovým zosilňovačom

Napätie na výstupe druhého stupňa pri meraní bolo 1,56 V, čo nie je až tak ďaleko od teoretickej hodnoty, pri bráni do úvahy viacero zanedbaní. Navyše pri konštrukcii sa používa viacero laserov, ktoré boli kúpené naraz. Je to kvôli tomu, že bolo vyrobených niekoľko prototypov a pomocných zariadení na testovanie a každé má svoj vlastný laser (strnosť sa merala iným laserom ako spomínaný priebeh). Medzi týmito lasermi môže

rozličný výkon, poprípade nemusia byť napájané presne rovnakým prúdom. Takže hodnota 1,56 V je relatívne blízko teoretickej. Toto napätie sa privádza na delič napätia $R1$ a $R2$, ako je na schéme 4.11. Striedavý signál sa skratuje z odporu $R2$ kondenzátorom $C5$, preto je na invertujúcom vstupe zosilňovača jednosmerné napätie pracovného bodu tranzistoru $Q2$ znížené o delič. Jednosmerný pracovný bod $Q2$ je

$$U_{Q2-DC-OUT} = V_{CC} - U_D - R_C \cdot I_C, \quad (4.33)$$

$$U_{Q2-DC-OUT} = 3,3 - 0,6 - 680 \cdot 1,5 \cdot 10^{-3} = 1,68 \text{ V}.$$

Napätie U_D je úbytok na dióde. Na invertujúcom vstupe bude napätie

$$U_{INV-DC} = U_{Q2-DC-OUT} \cdot \frac{R_2}{R_2 + R_1}, \quad (4.34)$$

$$U_{INV-DC} = 1,68 \cdot \frac{100 \cdot 10^3}{100 \cdot 10^3 + 3,9 \cdot 10^3},$$

$$U_{INV-DC} \approx 1,62 \text{ V},$$

čiže jednosmerné napätie na neinvertujúcom vstupe je približne o 60 mV väčšie. Ak začne prichádzať striedavý signál s frekvenciou 2,27 MHz napätie na invertujúcom vstupe sa meniť nebude, pretože konštanta RC článku je

$$\tau_{RC} = R_1 || R_2 \cdot C_5 = \frac{3900 \cdot 10^5}{3900 + 10^5} \cdot 10^{-6} \approx 3,75 \text{ ms}, \quad (4.35)$$

čo je oveľa viac, ako čas polovice periódy prichádzajúceho signálu. Takže ak bude signál na výstupe tranzistora menší ako 60 mV v zápornej polvlně, operačný zosilňovač sa preklopí a náhle vybije kondenzátor $C6$ cez diódu $D3$. Striedavý signál z výstupu je zaťažený 3,9 k Ω , keďže kondenzátor $C5$ skratuje striedavý signál na zem. Preto amplitúda nebude rovnaká ako vypočítaných 820 mV, ale

$$U_{INV-AC} = U_{Q2-AC-OUT} \cdot \frac{R_1}{R_1 + R_C} = 0,82 \cdot \frac{3900}{3900 + 680} \approx 700 \text{ mV}, \quad (4.36)$$

čo je stále viac ako 60 mV, preto je zosilnený signál dostatočne veľký. V čase kladnej polvlny sa kondenzátor začne nabíjať cez rezistor $R3$, pokiaľ nepríde znovu záporná polvlna. Takže v uzle za diódou je napätie, ktoré už pripomína demodulovaný signál. S rozdielom, že nemá strmé hrany a na spodku má priebeh rampy. Signál je na obrázku 4.12. Kvôli presne definovaným hranám a napäťovým úrovniám je signál poslaný cez buffer. Bez strmých hrán by bolo náročné presne určovať frekvenciu a rozoznávať

START bity. Tým, že dióda $D3$ má úbytok napätia približne 0,6 V, nedokáže vybit' kondenzátor úplne a CMOS buffer by takto pracoval na hranici preklopenia. Preto je napät'ovo posunutý o to isté napätie diódou $D4$. Kondenzátor $C7$ zabraňuje striedavému napätiu, aby sa vytváralo na dióde pri spínaní bufferu. Buffer má hysterézny vstup, aby pri prechodoch medzi signálmi neboli zákmity. Tým, že je buffer posunutý o napätie diódy, bude jeho výstupné napätie posunuté o 0,6 V nahor. Táto zmena mikrokontroléru nevedí, pretože je pod maximálnou úrovňou nízkeho napätia digitálneho vstupu. Hodnota $R3$ a $C6$ bola vypočítaná diferenciálnou rovnicou. Keďže prúd kondenzátorom je derivácia napätia podľa času, dá sa pre napätie na kondenzátore napísať

$$\begin{aligned} u_C &= V_{CC} - R \cdot i_C, \\ u_C &= V_{CC} - R \cdot \frac{C \cdot \partial v_C}{\partial t}. \end{aligned} \quad (4.37)$$

Po úprave je zrejmé, že ide o lineárnu diferenciálnu rovnicu prvého stupňa

$$u'_C + \frac{u_C}{RC} = \frac{V_{CC}}{RC}. \quad (4.38)$$

Jej riešenie je v tvare

$$u_C(t) = c \cdot e^{-\frac{t}{RC}} + 3,3. \quad (4.39)$$

Prvá počiatková podmienka je, že v čase 0 má napätie hodnotu 0,6 V (úbytok na dióde $D3$), z toho je možné vypočítať konštantu c

$$0,6 = c \cdot e^{-\frac{0}{RC}} + 3,3 \rightarrow c = -2,7. \quad (4.40)$$

Druhá počiatková podmienka je napätie na kondenzátore po prejení zápornej polvlny. Musí to byť hodnota pod minimálnym napätím, ktorým sa buffer preklápa. Inak by buffer pri týchto „špičkách“ pulzoval. Čas po prejení zápornej polvlny je polovica periódy nosného signálu

$$t_L = \frac{1}{2 \cdot f_{rez}} = \frac{1}{2 \cdot 2,27 \cdot 10^6} \approx 220 \text{ ns}. \quad (4.41)$$

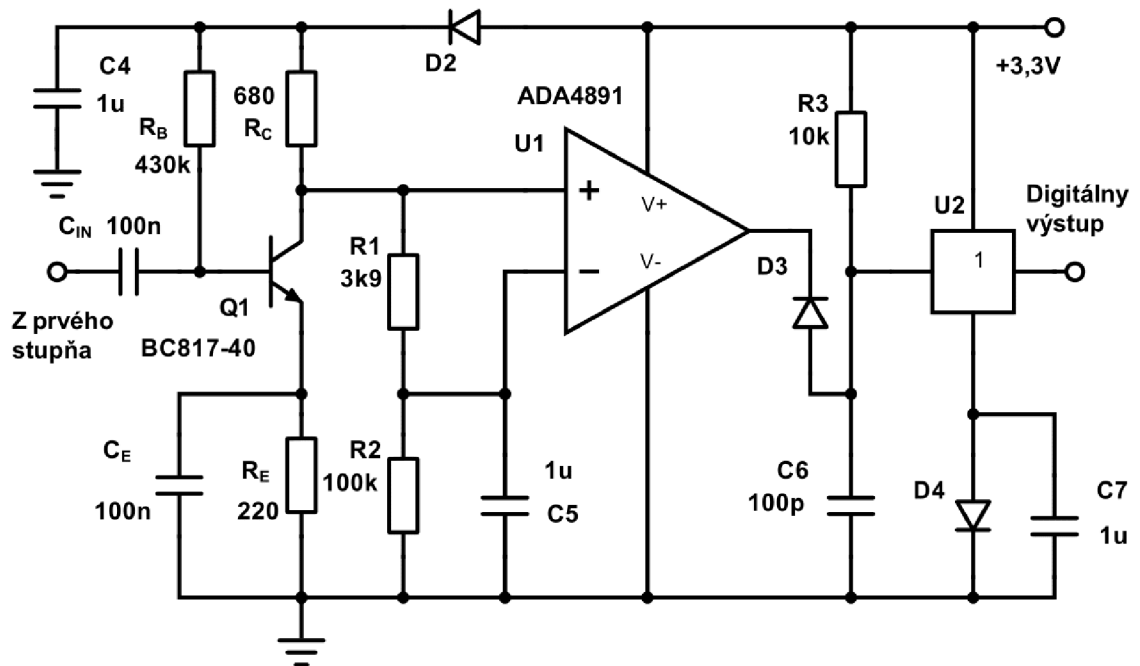
Buffer je integrovaný obvod SN74LVC1G17DBVR s minimálnou hodnotou $V_{IL} = 0,9$ V, od ktorej sa môže preklopiť pri 3 V napájaní. Netreba zabudnúť, že je posunutý o 0,6 V dohora diódou $D4$. Takže napätie, ktoré sa nesmie prekročiť je 1,5 V. Kvôli rezerve sa bude počítať s 1,2 V. Z toho je druhá počiatková podmienka

$$u_C(220 \text{ ns}) = 1,2 \text{ V.} \quad (4.42)$$

A teraz k výpočtu RC členu. Odpor bude zvolený na 10 kΩ, Úpravou sa potom dostane vzťah pre kondenzátor C v schéme označený ako C6

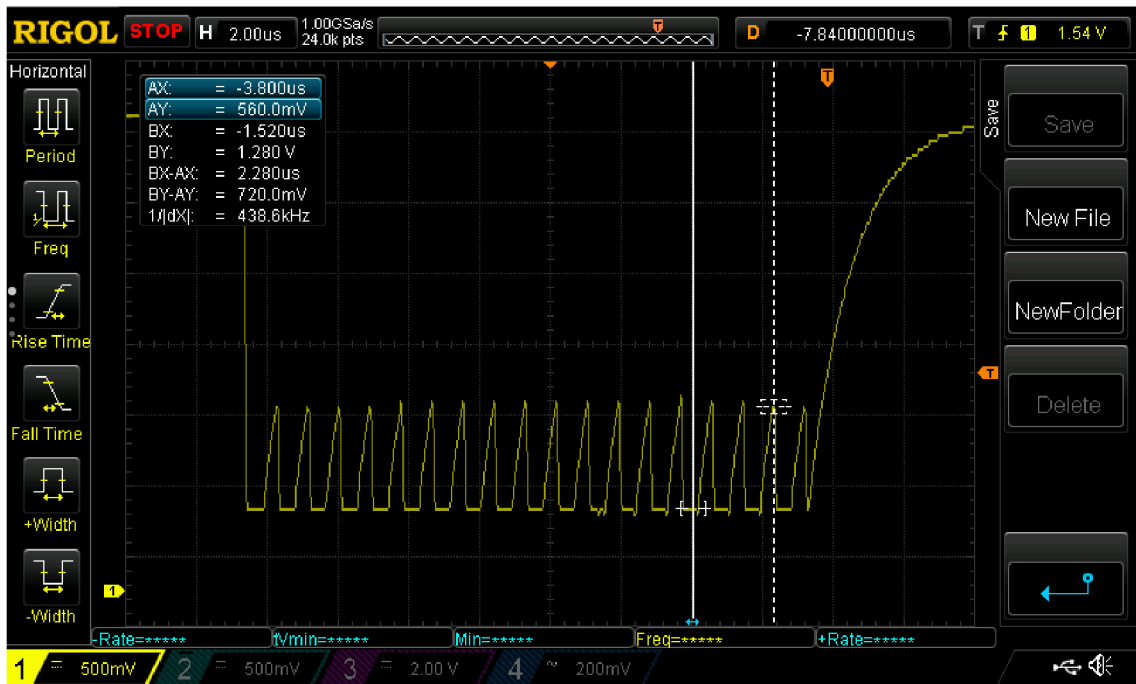
$$C_6 = -\frac{t}{R \cdot \ln\left(\frac{v_C - 3,3}{-2,7}\right)} = -\frac{220 \cdot 10^{-9}}{10 \cdot 10^3 \cdot \ln\left(\frac{1,2 - 3,3}{-2,7}\right)} = 87,5 \text{ pF.} \quad (4.43)$$

Hodnota kondenzátoru bude zaokrúhlená na 100 pF.



Obrázok 4.11 Druhá časť demodulátora

Všetky integrované obvody majú blokovací kondenzátor 100 nF, ktorý v schéme nie je uvedený, použité diódy sú 1N4148W. Buffer disponuje hysteréznym vstupom. Bez neho by sa mohol výstup preklopiť viackrát pri zmenách logických úrovní, kým by sa ustálil. Celá schéma aj s návrhom dosky plošného spoja sa nachádza v prílohe.



Obrázok 4.12 Tvar signálu pred bufferom

Keďže frekvencia 2,27 MHz vyhovuje dizajnu, môže sa vypočítať maximálna teoretická prenosová rýchlosť. Aby sa signál dal spracovať, musí sa stihnúť nakmitať na rezonančnom obvode, ako je na obrázku 4.10. Pre spoľahlivosť prenosu bude dĺžka jedného bitu približne 20 periód nosnej frekvencie. Najväčšia rýchlosť sa dosiahne pri prenose maximálnych 32 bajtov v jednom pakete. Prenosová rýchlosť nie je počet všetkých bitov, ktoré sa prenesú. Paket obsahuje START, LEN, RED a STOP bity, ktoré sa nepočítajú do efektívnej prenosovej rýchlosti. Potrebná dĺžka STOP bitov (bity tvoriace časovú medzeru medzi paketmi, aby mikrokontrolér mal čas odoslať dáta do PC) bola zistená experimentálne. Čas prenosu 32 bajtov s baud rate 115200 bol približne 75 μ s. Kvôli spoľahlivosti budú STOP bity dlhé 2 krát toľko, čiže 150 μ s medzi paketmi. START bity sú kratšie, ale je ich iba 8, takže ich časový rozdiel sa môže zanedbať. Rýchlosť bude vypočítaná pre paritu, ktorá obsahuje LEN bity a má 32 bitový redundanciu. Celý počet bitov v pakete bez STOP bitov je

$$\begin{aligned} PACKET_{bit} &= START + LEN + DATA + RED, \\ PACKET_{bit} &= 8 + 5 + 256 + 32 = 301 \text{ bitov.} \end{aligned} \quad (4.44)$$

Jeden bit trvá spomínaných 20 periód nosnej frekvencie, 301 bitov dáva čas

$$T_{packet} = 20 \cdot \frac{1}{f_{rez}} \cdot PACKET_{bit} = 20 \cdot \frac{1}{2,27 \cdot 10^6} \cdot 301 \approx 2,65 \text{ ms.} \quad (4.45)$$

Po pripočítaní časovej medzery 150 μ s vychádza na jeden paket 2,8 ms. Za tento čas sa preniesie 256 bitov. Teoretická prenosová rýchlosť tohto systému teda je

$$\text{Bit Rate} = 256 \cdot \frac{1}{T_{\text{packet}} + T_{\text{STOP}}} = 256 \cdot \frac{1}{2,8 \cdot 10^{-3}} \approx 91 \text{ kb/s}, \quad (4.46)$$

alebo 11,4 kB/s, čo spĺňa minimálne prenosové požiadavky. Presná rýchlosť prenosu sa zistí až meraním.

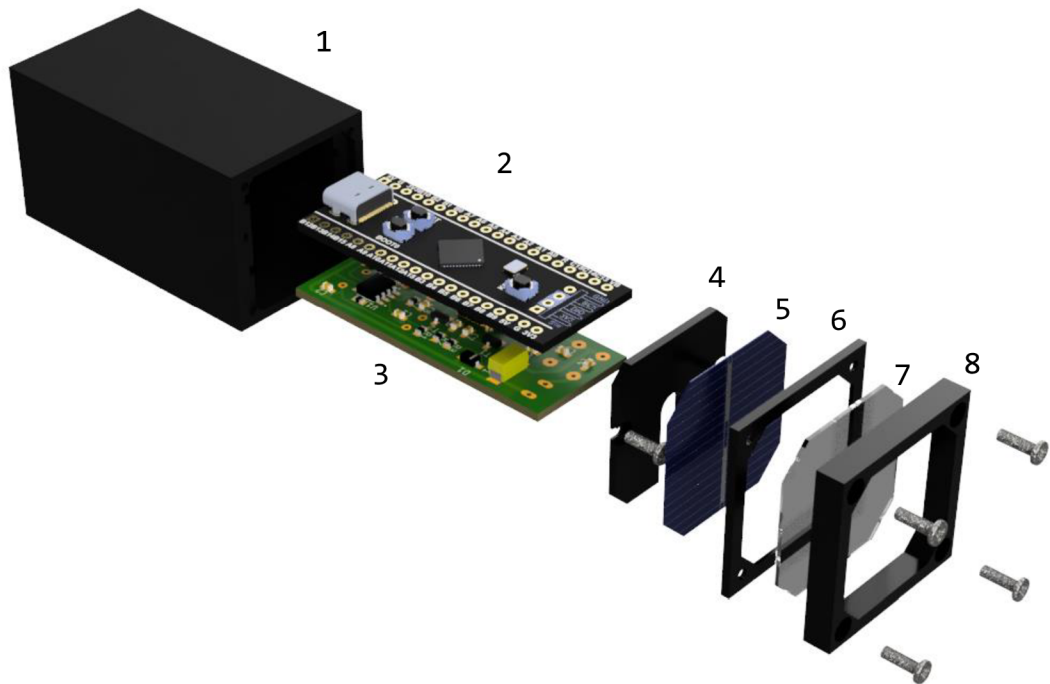
4.5 Krabičky

Skrinky vysielajú aj prijímača majú rovnaký rozmer, ktorý vychádza z použitého SoC (System on Chip) mikrokontroléru s DPS od firmy STMicroelectronics. Jeho presný typ, funkcie a programovanie bude v neskoršej kapitole. Rozmery DPS, na ktorej je mikrokontrolér prispájkovaný sú 21 x 53 mm, s hrúbkou 1,7 mm. Elektronika modulátora a demodulátora sa musí zmestiť do rovnakého rozmeru, aby mohli byť obe dosky zároveň nad sebou a lícovali po oboch stranách v krabičke. Dosky v krabičke nie sú priamo priskrutkované, ale zasúvajú sa do výrezov. Po zasunutí ich dotlačí vnútorná krytka. Tá sa prikrúca po oboch stranách, čím zaistí obidve dosky. Spredu je krabička štvorcová. Jej rozmery sa určili na 30 x 30 x 63 mm. Krabičky sú vytlačené na 3D tlačiarňami. Použitý je PLA plast.

V oboch krabičkách boli nad miestami tlačidiel RESET a BOOT dosky mikrokontroléru vyvrtané dierky. Je to kvôli tomu, aby aj po zložení prijímača a vysieláča bolo možné mikrokontroléry naprogramovať a to by bez prístupu k týmto tlačidlám nebolo možné.

4.5.1 Krabička prijímača

Skrinka prijímača je o niečo komplikovanejšia ako skrinka vysieláča a to z toho dôvodu, že je vpredu krehký solárny panel. Preto musí byť navrhnutá tak, aby bol panel na svojom mieste a zároveň aby na neho nič netlačilo, inak by mohol prasknúť. Už orezaný panel má rozmery 28 x 28 mm, s hrúbkou 0,22 mm. V súvislej vrstve na spodku a mriežkovito z vrchu, je na ňom naparená kovová vrstva. Táto vrstva je veľmi tenká, preto je potrebné dávať pozor pri spájkovaní prívodných vodičov, aby sa nerozpustila do spájky. Vhodné je na toto spájkovanie použiť olovnatú spájku.



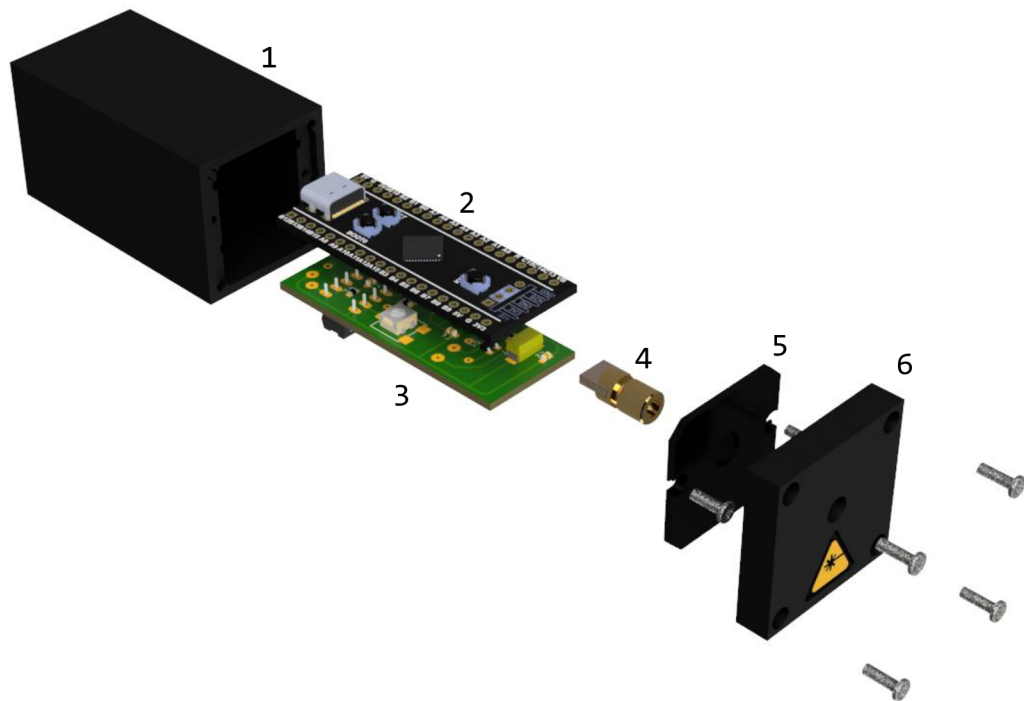
Obrázok 4.13 Konštrukcia prijímača, 1 telo krabičky, 2 mikrokontrolér, 3 demodulátor, 4 vnútorná krytka, 5 solárny panel, 6 ochranný rámček, 7 plexisklo, 8 predná krytka

Doska mikrokontroléru je s doskou demodulátora spojená vodičmi s plným jadrom, aby obe dosky držali pevne tvar. Tento celok sa zasúva do krabičky (do jej drážok), na ktorej je z druhej strany výrez pre USB C konektor a pritlačí sa priskrutkovaním plastovej vnútornej krytky. Vo vnútornej krytke je otvor na káblíky od panelu, ktoré idú k demodulátoru. Tak isto sú v nej zahĺbené otvory na skrutky a v krabičke je výrez na vnútornú krytku. Takže po jej priskrutkovaní je spredu rovná plocha. Na toto ide solárny panel, ktorý sa zakryje ochranným rámčekom. V rámčeku je 0,8 mm výrez na solárny panel, ktorý má hrúbku iba 0,22 mm. Preto je v ňom panel voľne uložený. Rámček zároveň zabezpečuje svojou hrúbkou aj vzduchovú medzeru medzi panelom a plexisklom, aby zatlačením spredu na prijímač panel nepraskol. Nakoniec ide predná krytka so štyrmi skrutkami v rohoch.

4.5.2 Krabička vysielača

Skrinka vysielača je konštrukčne jednoduchšia, laser na svojom mieste držia tvrdé prírodné vodiče (fotky v prílohe na konci). Presný smer určuje vnútorná krytka s vodiacou dierkou v strede. Telo krabičky je riešené rovnako, ako má prijímač. Takisto sú obe dosky spojené medzi sebou vodičmi s plným jadrom a vsúvajú sa do výrezov v tele

krabičky, kde ich vnútorná krytka drží na mieste. Vysielač má rovnako otvor na USB C konektor.



Obrázok 4.14 Konštrukcia vysielača, 1 telo krabičky, 2 mikrokontrolér, 3 modulátor, 4 laserová dióda, 5 vnútorná krytka, 6 predná krytka

Keďže ide o čierny PLA plast je nebezpečné ho nechávať na slnku, alebo na miestach s vyššími teplotami. Tento plast začína mäknúť už od teploty 70 °C a mohol by sa deformovať [24]. Najpodstatnejšie parametre zo slicera (nastavenie 3D tlače) sú v tabuľke 4.1. Priľnavosť k podložke bude zvýšená rozšíreným okrajom prvej vrstvy (z angličtiny „brim“). Použitá tlačiareň bude Creality Ender 3.

Tabuľka 4.1 Parametre 3D tlače

parameter	hodnota
výška jednej vrstvy	0,12 mm
priemer trysky	0,4 mm
hrúbka steny	1,2 mm
percento výplne	30%
teplota trysky	200 °C
rýchlosť tlače	50 mm/s
rýchlosť tlače prvej vrstvy	10 mm/s
teplota podložky	50 °C

Všetky 3D diely aj kompletná konštrukcia prijímača a vysielača (ako sú na obrázku 4.13 a 4.14) sú v prílohe, vyexportované z návrhového prostredia Fusion 360.

5 REALIZÁCIA SOFTVÉROVEJ ČASTI

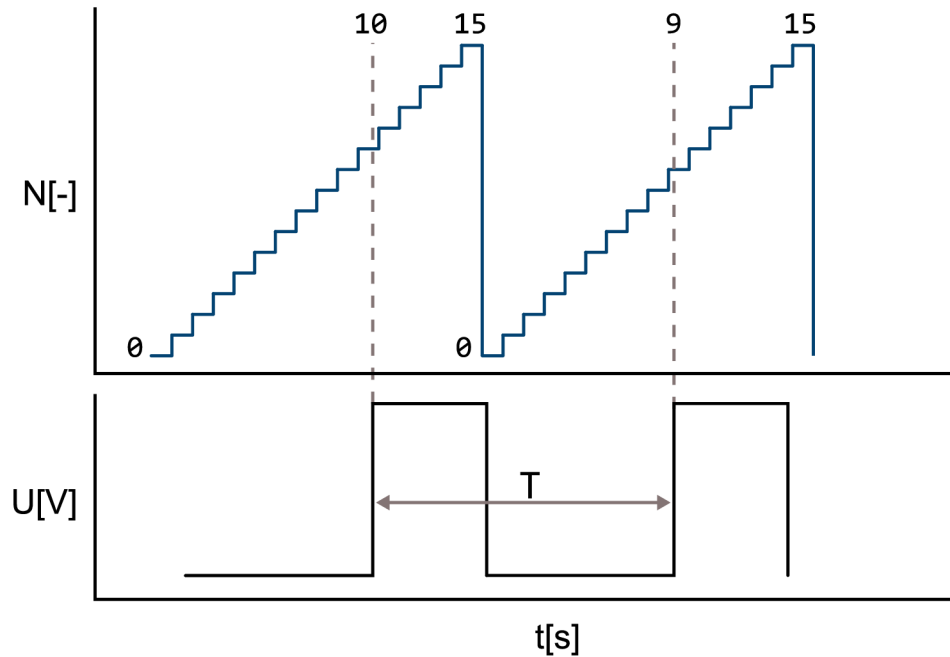
Pred implementáciou jednotlivých algoritmov na odhalenie chýb bude najskôr prebraté, ako správne rozoznať začiatok paketu. Zčať vzorkovať v správnom čase je kritické a bez správne načasovaného vzorkovania sú dáta nepoužiteľné. Použitý mikrokontrolér vo vysieláči je STM32F401CC a v prijímači STM32F411CE³. Mali byť rovnaké, ale žiaľ predajca urobil chybu a odoslal dva rôzne. Takto je náročnejšie nastaviť na obidvoch hardvérových časovačoch rovnakú periódu. Ako bolo navrhnuté v podkapitole Synchronizácia, začiatok paketu indikuje zrýchlené striedanie logickej úrovne. Dalo by sa povedať, že prenosový systém pracuje s dvoma typmi modulácie. Prvá frekvenčná, na rozoznanie začiatku paketu a druhá amplitúdová, pre jednotlivé bity paketu.

5.1 Signálová vrstva

V tomto prípade sa začne najskôr prijímačom a potom vysieláčom. Najdôležitejšie pre signálovú vrstvu je spoľahlivé nájdenie začiatku paketu. Ako vždy existuje viacero možností ako rozoznať, že je na vstupe iná frekvencia. Ale spravidla (a hlavne ak ide o mikrokontroléry), najlepšia je tá, ktorá využíva hardvérové vybavenie. Prídavný hardvér pracuje takmer, alebo úplne bez pomoci procesora (takže mu nekradne výpočtový čas) a oveľa rýchlejšie. Jedným z týchto hardvérových vybavení sú časovače (timers), ktoré je možné nakonfigurovať do ICM (z angličtiny „*Input Capture Mode*“). Čo znamená, že ak signál na nastavenom vstupe zmení úroveň, hodnota počítadla časovača sa preniesie do záchytného registra a vyvolá sa prerušenie procesora. Tento mód sa používa na presné meranie frekvencie signálu. Samozrejme hodnotu zo záchytného registra je potrebné ukladať inak sa prepíše. V tomto prípade bude ICM nastavený na vzostupnú hranu prichádzajúceho signálu. Už po druhom prerušení je možné vypočítať frekvenciu z dvoch hodnôt počítadla. Ak časovač nepretiekol, odčíta sa aktuálna hodnota záchytného registru od predošlej (uloženej) a vynásobí sa časom jedného kroku počítadla. Takto sa získa čas jednej periódy. Rýchlosť, akou počítadlo zvyšuje svoju hodnotu, sa nastavuje deličkami, ktoré delia vnútornú frekvenciu čipu. Ak počítadlo pretieklo, tak k ich rozdielu je ešte potrebné pripočítať maximálnu hodnotu počítadla toľkokrát, koľko pretieklo. Potom sa výsledok vynásobí časom jedného kroku. V programe prijímača nie je potrebné zisťovať, aká frekvencia je na vstupe, ale iba či

³ [Odkaz na technický list](#)

chcená frekvencia je na vstupe. Takže v tomto prípade, ak počítadlo pretečie viac ako raz, uložená hodnota počítadla sa vynuluje a bude sa čakať na ďalšie prerušenia. Je preto potrebné nastaviť deličky tak, aby sa do času za ktorý počítadlo pretečie zmestil dvojnásobok času START bitu.



Obrázok 5.1 Input Capture Mode

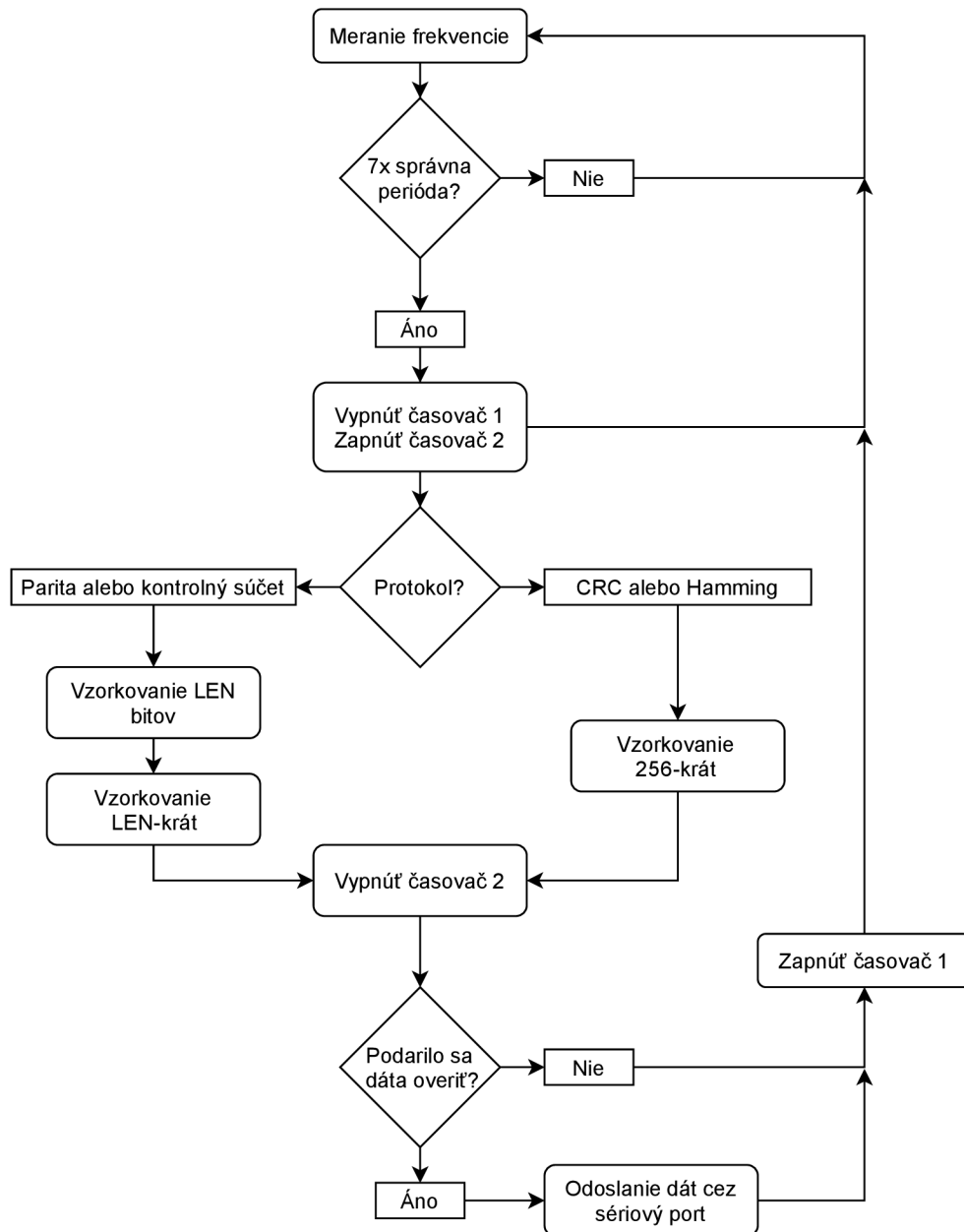
Na obrázku 5.1 je teoretický princíp ICM. Počítadlo počíta opakovane do 15. Pri prechode vstupného signálu z 0 do 1 sa hodnota počítadla uloží. Ak by jeden krok počítadla T_C trval 100 ms, frekvencia vstupného signálu by bola

$$f = \frac{1}{T_C \cdot (N_2 - N_1 + N_{MAX})} = \frac{1}{0,1 \cdot (9 - 10 + 15)} = 0,71 \text{ Hz.} \quad (5.1)$$

5.1.1 Prijímač

Vývojový diagram pre program prijímača je na obrázku 5.2. Meraná frekvencia na začiatku musí byť v správnom intervale 7-krát po sebe. Aj keď vysielateľ odošle 8 periód a teda v signále bude 8 nábežných hrán, perióda sa dá zistiť iba z času dvoch hrán medzi sebou a teda iba 7-krát. Ak je splnená táto podmienka, časovač merajúci vstupnú frekvenciu sa vypne a zapne sa druhý časovač. Ten bude čítať logickú úroveň v pravidelných intervaloch. Pri algoritmoch parita a kontrolný súčet sa najskôr prečíta 5 LEN bitov. Z nich sa vyhodnotí dĺžka DATA bitov. Hammingov kód a CRC majú vždy

256 DATA bitov, takže po START bitoch sa hneď začnú čítať DATA bity. Po vzorkovaní sa vypne druhý časovač a dáta sa vyhodnotia (v skutočnosti sa dáta vyhodnocujú priebežne pri vzorkovaní (okrem parity), ale v diagrame to je naznačené na konci, pretože to je čas kedy sa vyhodnocovanie skončí a prijímač vie či je správa poškodená alebo nie). Ak sú dáta platné, odošlú sa sériovým portom a celý algoritmus začína odznova.

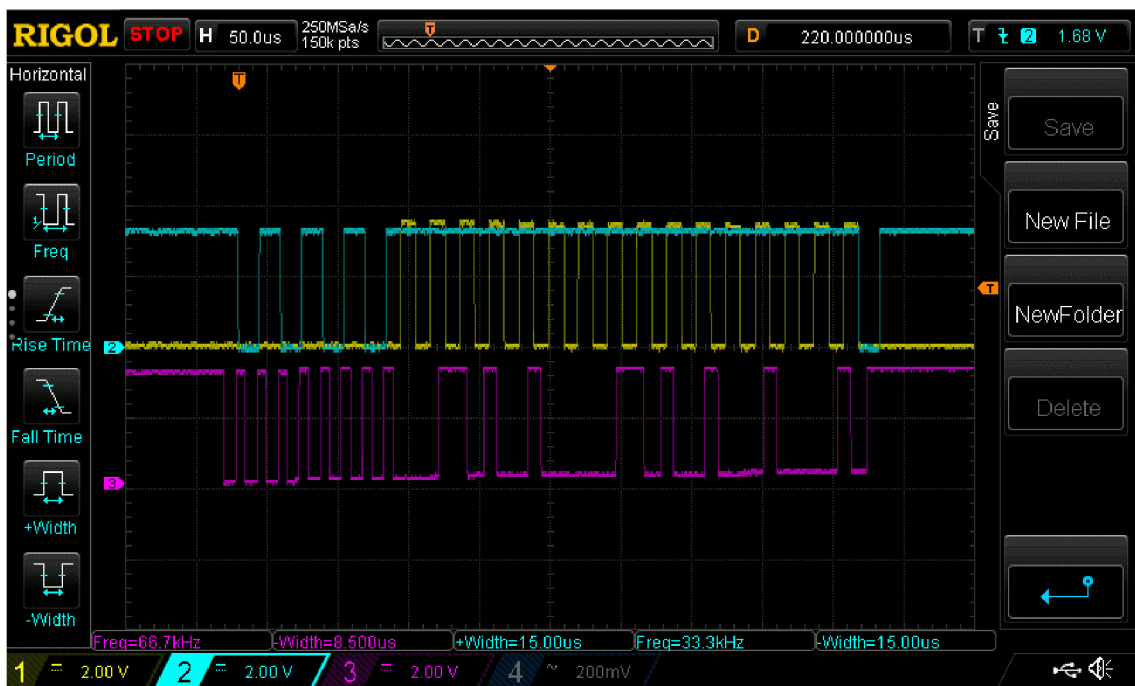


Obrázok 5.2 Vývojový diagram signálovej vrstvy prijímača

Keďže ide o sériový prenos, dáta pribúdajú po bitoch a prijímač ich skladá do bajtov. Signálová vrstva je robená tak, že ako prvý sa odosiela MSB (najvýznamnejší bit) a ako posledný LSB (najmenej významný bit). Pri práci s časovačmi je dobré nastaviť hodnotu počítadla, aby časovač nezačal rátať od náhodnej hodnoty. Pre časovač 1,

merajúci frekvenciu, na tom až tak nezáleží. Ale pre druhý časovač, vzorkujúci bity, to je kritické. Pred zapnutím sa jeho počítadlo naplní tak, aby vzorkoval približne do stredu prichádzajúcich bitov.

Pri hľadaní chýb je najlepšie vidieť pracovanie obvodu v reálnom čase. Obidva zariadenia sa programovali bez možnosti debuggeru. Sú ale aj iné spôsoby ako zistiť, čo sa v obvode deje. V tomto prípade sa na začiatok funkcie merajúcu frekvenciu pridalo makro, ktoré otočí úroveň napätia na výstupe (časovač 1, modrý kanál). Tak isto do vzorkovacej funkcie (časovač 2, žltý kanál). Takže vždy, keď sa zavolá jedna z funkcií, korešpondujúce napätie sa preklopí. Toto nastavenie pomohlo rýchlemu hľadaniu chýb v kóde. Na obrázku 5.3 je zachytená správa „Hi!“ overovaná paritou. Fialový priebeh je signál z demodulátora. Makro, ktoré mení modrý priebeh, bolo ešte pridané na miesto pred začatím a tak isto skončením odosielania dát do počítača, aby bolo jasné aká časová rezerva musí byť medzi paketmi.

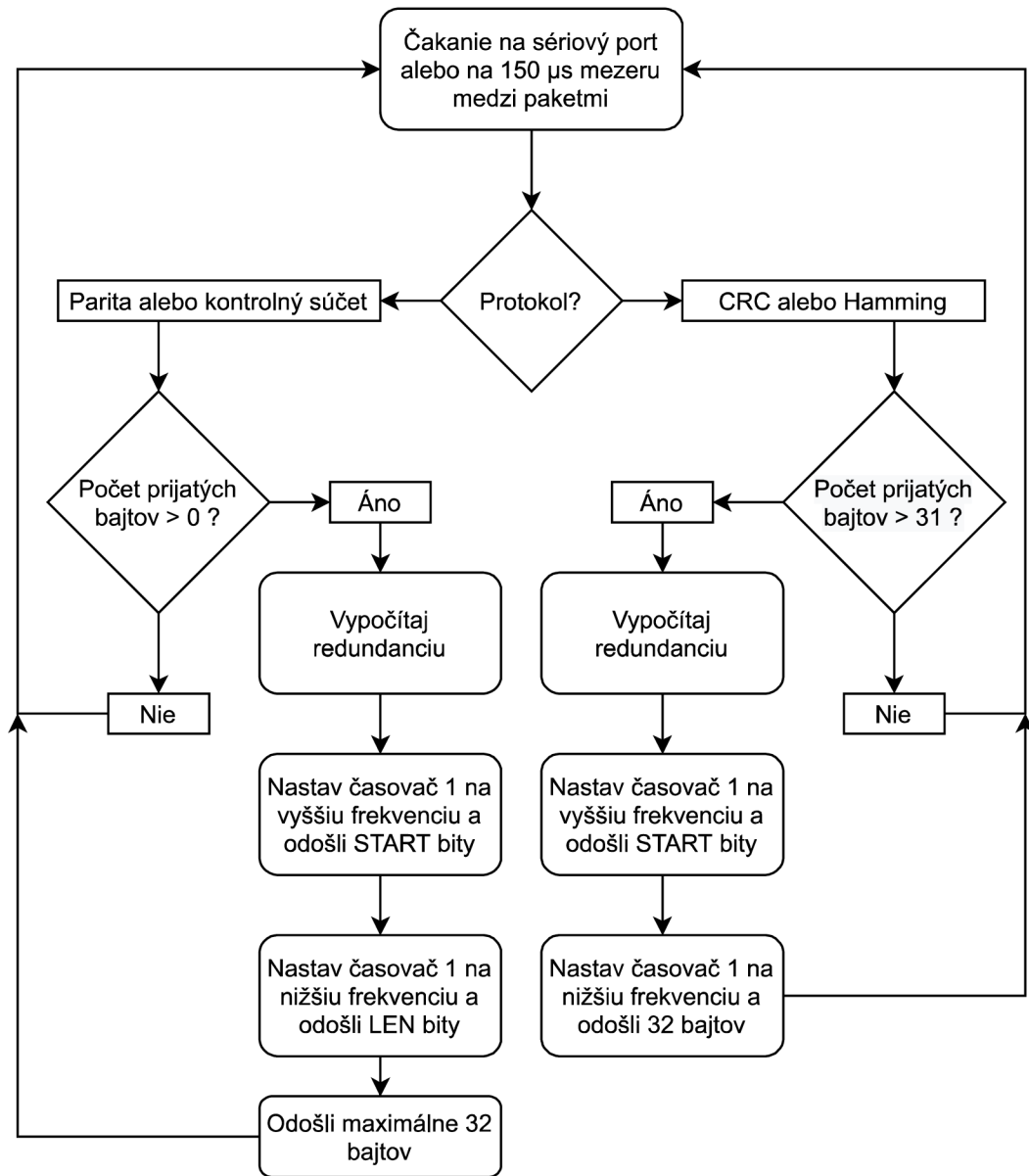


Obrázok 5.3 Debugovanie pomocou osciloskopu

5.1.2 Vysielač

Ak ide o vysielač, začiatok paketu s rýchlejšou frekvenciou, aj odosielanie jednotlivých bitov s pomalšou frekvenciou bude mať na starosti jeden časovač. Časovače sa dajú prekonfigurovať aj za behu. V použítom mikrokontroléri sa nastaví, že ak sa zadá nová hodnota, po ktorú má počítadlo počítať, aplikuje sa až po pretečení. Takto je možné

plynulo zmeniť časový interval výstupných bitov. Pretože bity sa do modulátoru odosielajú vo funkciách, ktorá sa volá pri pretečení. Vývojový diagram funkcie vysieláča je na obrázku 5.4. Vyrovňavacia pamäť sériového portu je 64 bajtov.



Obrázok 5.4 Vývojový diagram signálovej vrstvy vysieláča

Program pracuje s dvoma poľami bajtov. Jedno sa napĺňa a druhé sa odosiela. Po odoslaní sa ukazovatele na polia vymenia. Pre overenie paritou a kontrolným súčtom stačí aj jeden bajt na spustenie odoslania v prípade, že je vyrovňavacia pamäť prázdna. Druhý 2 algoritmy potrebujú 32 bajtov. Keď príde z počítača naraz 64 bitov, prijímač naplní pole do maxima (32 bajtov) a potom ho odošle. Pred odoslaním druhých 32 bajtov čaká 150 µs čím dáva čas prijímaču na odoslanie dát do počítača. Vysielač nebol pripojený na osciloskop, pretože jeho výstup je vidieť na výstupe demodulátora v prijímači.

5.2 Dátová vrstva

Signálová a dátová vrstva sú úzko späté, pretože sú implementované v tom istom kóde. Dátová vrstva sa zaoberá prijímaním, spracovaním a odosielaním dát. Jej hlavná úloha je generovanie redundancie na strane vysielača a overovanie dát na strane prijímača. Overovanie dát v prijímači sa deje po častiach, tak ako prichádzajú dáta (s výnimkou parity). Vo vysielači sa redundancia počíta po častiach pre všetky algoritmy.

5.2.1 Parita

Tento algoritmus je jednoduchý, rýchly, ale najmenej bezpečný. Overenie paritou v prijímači prebehne tak, že po navzorkovaní všetkých bitov sa nimi prejde cyklom. V ňom sa skontroluje, či každý deviaty bit je rovný parite predchádzajúcich ôsmich bytov. Parita má ako jediná redundanciu za každým bajtom zvlášť, tak ako je načrtnuté na obrázku 3.3. Ak overenie zlyhá, aj len pre jediný bajt v pakete, všetky dáta sa zahodia. Pre kontrolu parity bola použitá vstavaná GCC funkcia (2.3). Parita sa vo vysielači počíta priebežne pri posielaní bitov. Po odoslaní ôsmich bitov sa odošle aj vypočítaná parita.

5.2.2 Kontrolný súčet

Kontrolný súčet sčítava všetky prijaté bajty v prijímači tak, ako prichádzajú. Po prijatí posledného bitu sa vyhodnotí, či prijatý kontrolný súčet na konci dát odpovedá vypočítanému. Kontrolný súčet prenášajú 2 bajty aby sa do neho vošiel aj súčet tridsiatich dvoch maximálnych 8-bitových hodnôt. Dal by sa použiť aj jeden bajt s postupom vysvetleným v podkapitole Kontrolný súčet, ale to uberá z bezpečnosti a pridáva na náročnosti algoritmu, čo sa neoplatí vymieňať za uvoľnenie jedného bajtu. Je pravda, že tento algoritmus vie byť veľmi neefektívny pri prenose jedného bajtu, pretože redundancia v takom prípade zaberá 2-krát toľko miesta ako dáta, nehovoriac o START, LEN a STOP bitoch.

5.2.3 Hammingov kód

Hammingov kód je upravený tak, že prvý bajt začína s indexom 1 (nie 0) a redundancia sa posielajú na konci, nie v strede medzi dátami. Nultá pozícia sa nepoužíva. Ide o rovnaký princíp ako bol vysvetlený v teoretickej časti. Vysielač posielajú dáta po bitoch a zároveň počíta pozíciu každého bitu. Ak je posielaný bit logická jednotka, vysielač spraví XOR jeho indexu s výsledkom predchádzajúcich XOR operácií. Táto XOR premenná je na začiatku prenosu každého paketu inicializovaná na 0. Po poslednom DATA bajte sa

odošle aj samotná XOR premenná. Prijímač opakuje rovnaké kroky a po prijatí XOR premennej vykoná XOR aj s ňou. Ak sú dáta platné, výsledok tejto operácie je 0. Pretože je index posunutý o jedna, 256 indexov sa nezmestí do ôsmich bitov, ale až do deviatich. Preto sa posiela XOR premenná v dvoch 8 bitových hodnotách, čo je kompromis tejto modifikácie algoritmu.

5.2.4 Cyklický redundantný súčet

Aj keď je CRC najkomplikovanejší algoritmus, bolo ho najjednoduchšie implementovať, keďže všetky mikrokontroléry z rodiny STM32Fxxx majú dedikované hardvérové vybavenie na počítanie CRC. V prípade použitých mikrokontrolérov STM32F4xx je situácia ešte jednoduchšia, pretože nie je možné nič nastaviť [32]. Čo je na druhú stranu škoda, pretože nie je možné nastaviť ani deliaci polynóm. To však v tomto prípade nevádi. Prijímač a vysielateľ majú rovnaký polynóm a to CRC-32 Ethernet polynóm

$$dp(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1. \quad (5.2)$$

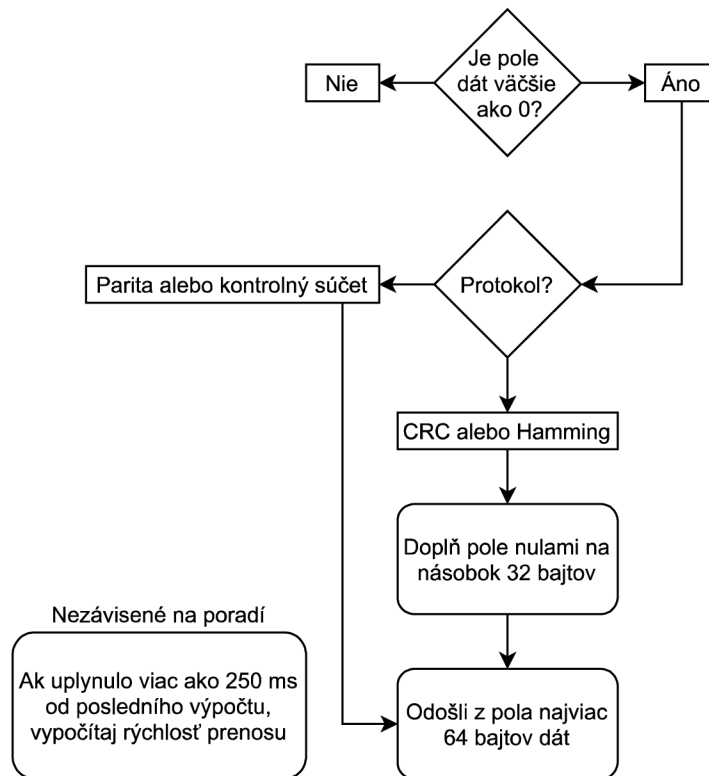
Tento polynóm má pre správu do 268 bitov HD = 8 [30]. Pred jeho použitím sa mu musí povoliť prístup k hodinovému signálu. Hardvérové vybavenie sa obsluhuje iba cez 2 registre. Do prvého sa vloží 32 bitová hodnota pre ktorú sa CRC počíta. Ak sa CRC počíta pre dlhšiu správu, tak sa do tohto registra vkladá postupne. Vždy ale až po skončení výpočtu, ktorý trvá 1287 hodinových cyklov [36]. Druhý register je na nastavenie periférie. Odomknutý je v ňom iba jeden bit. Ten resetuje CRC hardvér. Inicializačná hodnota po resete je 0xFFFFFFFF. Konkrétne ide o verziu CRC-32/MPEG-2. Parametre sú zhrnuté v tabuľke 5.1. Po skončení výpočtu je výsledok v rovnakom registri.

Tabuľka 5.1 Parametre pre CRC-32/MPEG-2

parameter	hodnota
Check	0x0376E6E7
Poly	0x04C11DB7
Init	0xFFFFFFFF
RefIn	false
RefOut	false
XorOut	0x00000000

5.3 Transportná vrstva

Transportná vrstva zabezpečuje prenos dát medzi počítačom a mikrokontrolérom. Táto vrstva je implementovaná v prehliaďači. V podkapitole Prenos bolo zmienené, že dáta do tejto vrstvy je možné vložiť aj priamo, avšak bez záruky, že dorazia celé, ak sa nevojdú do jedného paketu. Na strane vysielača je dáta možné zadať cez textové pole, alebo zo súboru. Na obrázku 5.5 je vývojový diagram transportnej vrstvy vysielača.



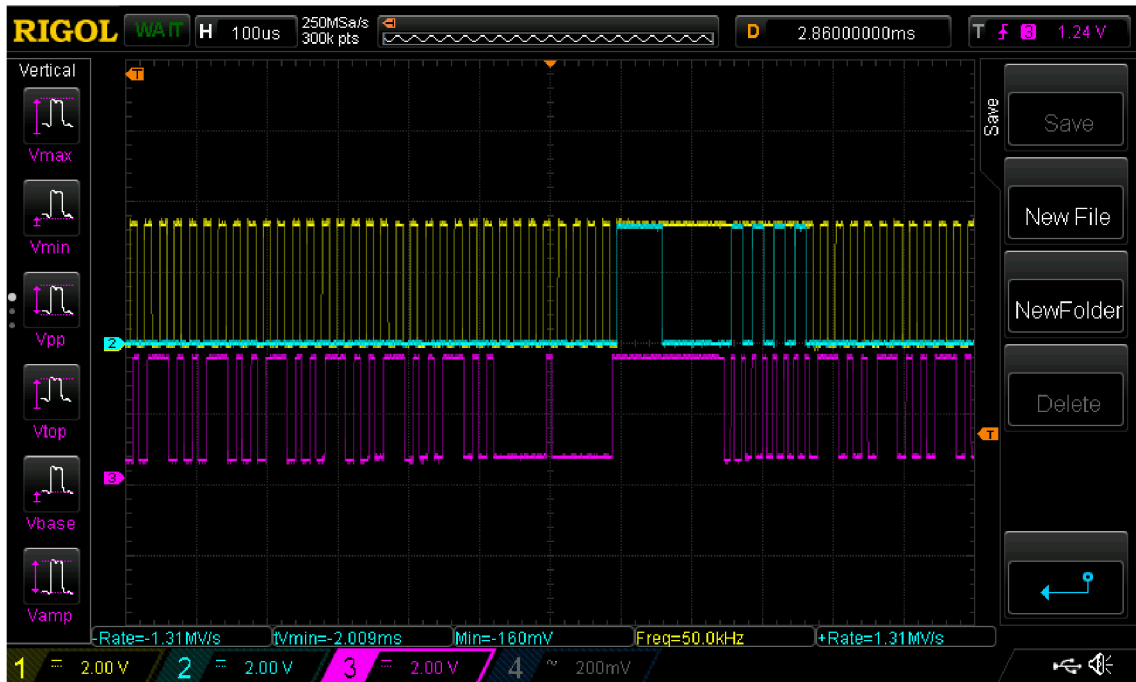
Obrázok 5.5 Vývojový diagram transportnej vrstvy vysielača

Pole dát sa naplní ihneď celým objemom vložených dát po stlačení tlačidla odoslať. Funkcia popísaná vývojovým diagramom sa volá približne v 7 ms intervaloch. Na jeden raz sa odosiela maximálne 64 bajtov dát (2 pakety), pretože mikrokontrolér má 64 bajtovú vyrovnávaciu pamäť pre sériový port. Takže priemerný čas na jeden paket je 3,5 ms. Je dôležité, aby bol tento čas väčší, ako čas potrebný k preneseniu jedného paketu (4.45) zvýšený o čakaciu rezervu 150 μ s signálovej vrstvy. Čo je spolu 2,8 ms. Rýchlosť pre 7 ms interval je

$$\text{Bit Rate} = \frac{N_{BITE}}{t_{BITE}} = \frac{64}{7 \cdot 10^{-3}} \approx 9143 \text{ B/s.} \quad (5.3)$$

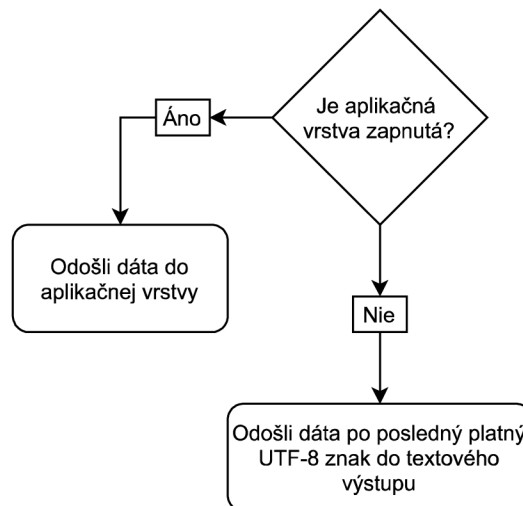
Teoreticky by bolo možné volať spomínanú funkciu aj v 6 ms intervaloch. Pretože 3 ms je stále menej ako 2,8 ms. Pri testovaní sa však stávalo, že vyrovnávacia pamäť po čase

pretiekla. Obrázok 5.6 je zameraný na 150 μ s medzeru medzi paketmi. V čase logickej jednotky modrého priebehu sa dáta odosielať do počítača. Za ňou (čas logickej nuly) nasleduje rezerva a potom START bity.



Obrázok 5.6 Medzera medzi paketmi

Funkcia transportnej vrstvy je posielat' dáta do aplikačnej vrstvy, alebo do textového výstupu, ak je aplikačná vrstva vypnutá. Na obrázku 5.7 je vývojový diagram transportnej vrstvy prijímača. Tá sa zavolá vždy ak dorazia dáta.



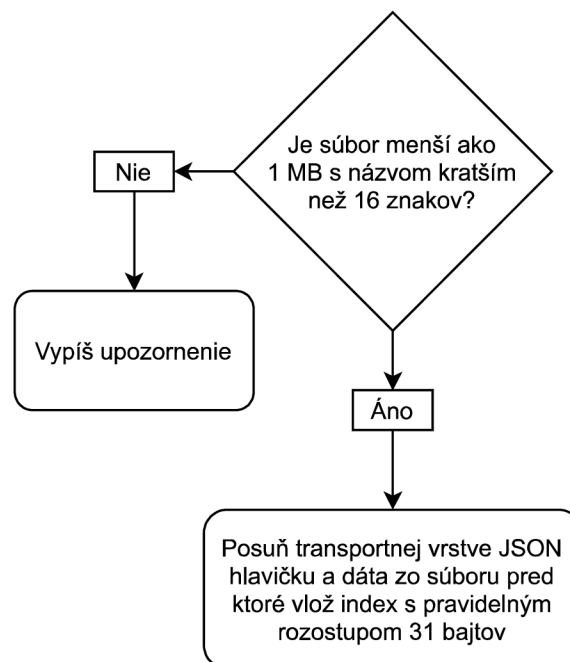
Obrázok 5.7 Vývojový diagram transportnej vrstvy prijímača

Pri posielaní do textového výstupu je potrebné kontrolovať posledný platný UTF-8 znak. Prijímač nepozná koľko dát má doraziť a častokrát dáta nedorazia v celku,

ale po častiach (môže za to oscilácia rýchlosti dát, ktorá je vysvetlená na konci podkapitoly Grafické rozhranie). Ak by sa prenášali iba ASCII symboly, neznamenaloby to žiadny problém. V tomto prípade sa ale prenášajú UTF-8 symboly, ktoré môžu mať až 4 bajty. Takže môže nastať situácia, že sa posiela 4 bajtový UTF-8 znak, ktorý sa prijme na 2-krát po dva bajty. Ak by sa do textového výstupu vypisovalo všetko tak ako prichádza, pribudli by namiesto správneho znaku dva neznáme znaky (označené napríklad symbolmi ❖❖). Preto sa dáta môžu odoberať a vypisovať len po posledný platný znak.

5.4 Aplikačná vrstva

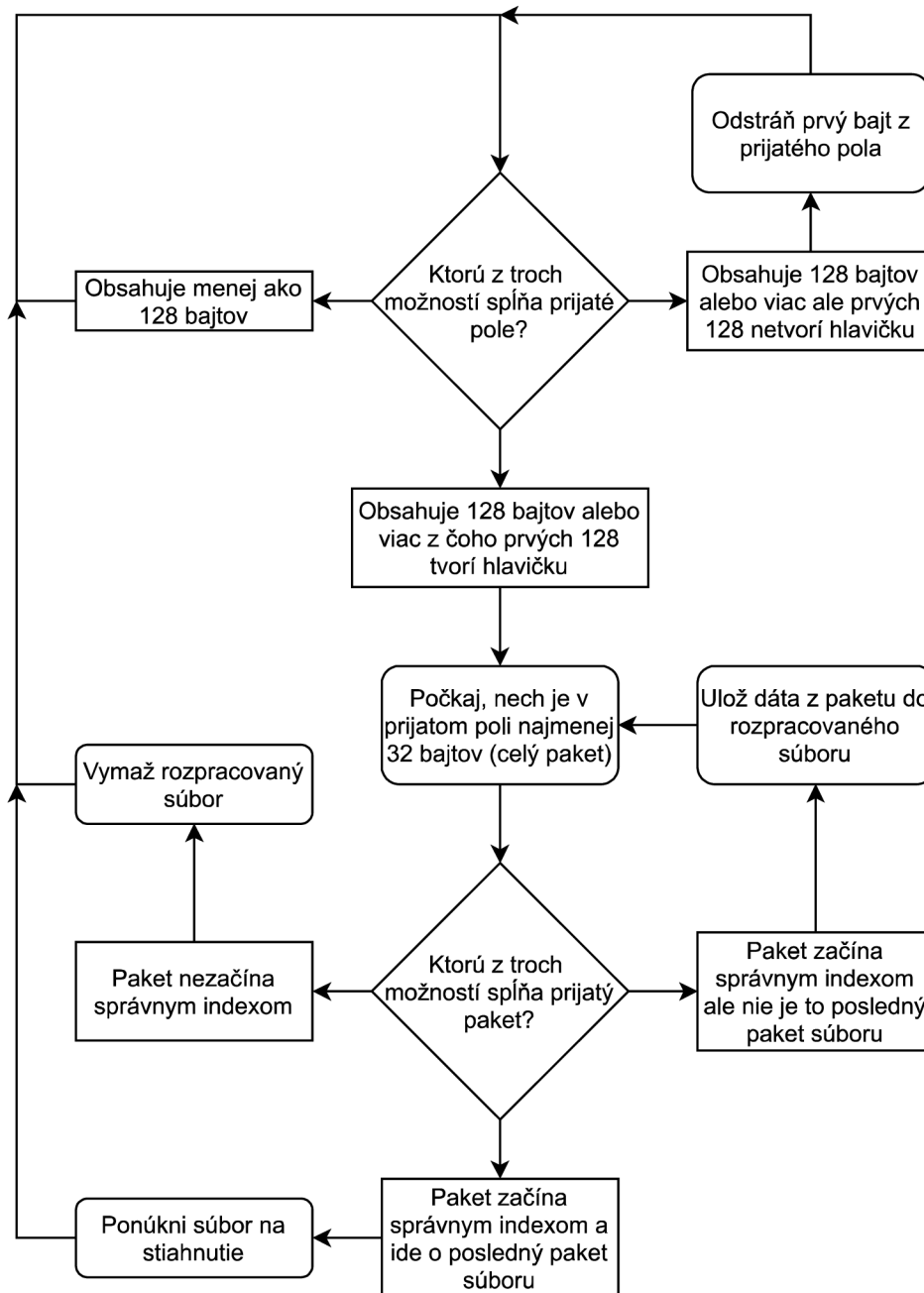
Aplikačná vrstva využíva transportnú vrstvu na prenos súborov. Na strane vysielajúca rozdelí dáta na 31 bitové časti, pred ktoré zaradi indexy jednotlivých paketov. Okrem toho vytvorí aj JSON hlavičku s fixnou dĺžkou 128 bitov, ktorú posunie transportnej vrstve ako prvú. Na obrázku 5.8 je vývojový diagram aplikačnej vrstvy vysielajúca. Jeho funkcia sa volá vždy, ak užívateľ vyberie súbor a stlačí tlačidlo odoslať. Ak je vybraný súbor väčší ako 1 MB, alebo je jeho názov dlhší ako 16 znakov, v prehliadači sa zobrazí upozornenie a súbor sa neodošle.



Obrázok 5.8 Vývojový diagram aplikačnej vrstvy vysielajúca

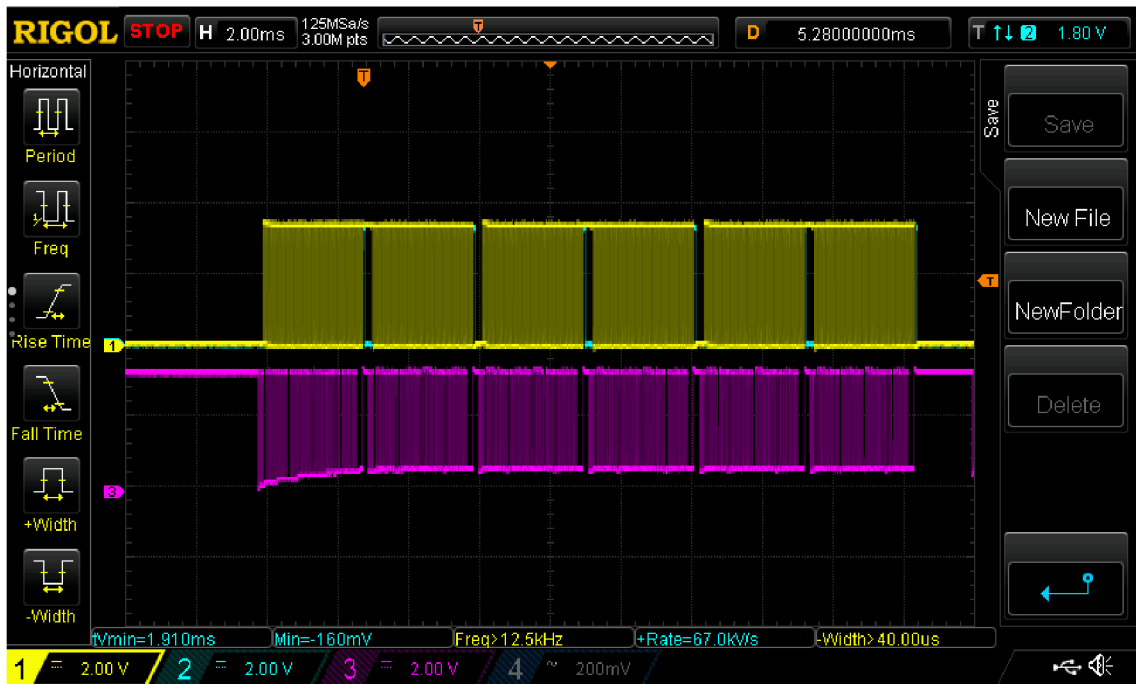
Na strane prijímača vykonáva aplikačná vrstva opačné kroky. Skontroluje hlavičku a potom index v každom pakete. Takto overí, že sa súbor preniesol celý a nechýba z neho

žiadna časť. Nazvaná bola FILE protokol, pretože zabezpečuje prenos súborov. Ak hlavička nebude mať správny formát, alebo z nej bude chýbať paket, súbor sa zahodí. Aplikačná vrstva potom znovu čaká na hlavičku. Na obrázku 5.9 je vývojový diagram aplikačnej vrstvy prijímača. Pole s názvom prijaté pole priebežne plní transportná vrstva.



Obrázok 5.9 Vývojový diagram aplikačnej vrstvy prijímača

Na obrázku 5.10 je prenos šiestich 32 bajtových paketov. Ide o 62 bajtový súbor. Prvé 4 pakety obsahujú JSON hlavičku a druhé 2 dáta zo súboru. Medzery po každých 2 paketoch (64 bajtov) sú o niečo širšie, kvôli 7 ms intervalom transportnej vrstvy.



Obrázok 5.10 Prenos súboru

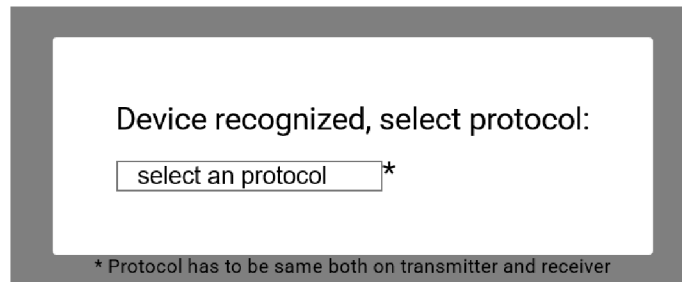
5.5 Grafické rozhranie

Grafické rozhranie uľahčuje prácu užívateľom. Ak ide o jednoduchý program, dobre spravené grafické rozhranie je intuitívne a užívatelia ho dokážu obsluhovať aj bez návodu. V tomto prípade je kompletná aplikácia v jedinom HTML súbore, ktorý sa nakonfiguruje podľa pripojeného zariadenia. Stránka pred konkrétnou konfiguráciou je na obrázku 5.11. Prenos súborov bol povolený len s pokročilejšími algoritmi, CRC a Hammingov kód, ale fungoval by aj s paritou, alebo kontrolným súčtom.

<input type="button" value="Connect"/> <input type="button" value="Disconnect"/>	
Status:	Disconnected
Device:	-
Protocol:	-
Enable FILE protocol <input type="checkbox"/> <small>Can be used only with CRC and Hamming code</small>	

Obrázok 5.11 Úvodná stránka

Po stlačení tlačidla Connect a vybratí sériového portu sa zobrazí okno, v ktorom užívateľ musí vybrať algoritmus, aby mohol pokračovať ďalej. Okno s výberom algoritmu je na obrázku 5.12.



Obrázok 5.12 Výber protokolu

Pozadie stránky stmavne, čo naznačuje nutnosť výberu. Vybratý algoritmus vyvolá odoslanie jedného z reťazcov sériovým portom. Reťazce pre jednotlivé algoritmy sú vypísané v tabuľke 3.1.

Connect
Disconnect

Status:	Sending FILE <div style="width: 50px; height: 10px; background: linear-gradient(to right, blue, gray);"></div>
Device:	Transmitter-STM32F401CC
Protocol:	Cyclic redundancy check
Speed:	9360 B/s
Top speed:	9434 B/s

Enable FILE protocol

Can be used only with CRC and Hamming code

Text Input

Hello world! 😊😊 (づ_づ)づ

File Input


No file chosen

Obrázok 5.13 Grafické rozhranie pre vysielateľ

Keďže zvolením sériového portu prehliadač vie, aký mikrokontrolér s ním komunikuje, po vybratí algoritmu sa v prehliadači zobrazia odpovedajúce elementy. Ak je pripojené zariadenie vysielateľ, stránka sa zmení na takú, ako je na obrázku 5.13.

V druhom prípade ide o prijímač a stránka bude vyzerat' ako na obrázku 5.14. Obe majú textové pole na zadávanie a čítanie textových reťazcov a miesto kde sa nahrávajú a sťahujú súbory. Obrázky boli spravené po prenesení jedného UTF-8 reťazca, jedného súboru s názvom test.bmp a počas prenosu druhého súboru. Keď sa prenášajú dáta, tlačítka na odosielanie a voľba protokolu na prenos súborov sú uzamknuté.

Connect Disconnect

Status: Receiving FILE 

Device: Receiver-STM32F411CE

Protocol: Cyclic redundancy check

Speed: 9143 B/s

Top speed: 21669 B/s

Enable FILE protocol
Can be used only with CRC and Hamming code

Text Output

Hello world! 😊😊 (づ_づ)づ

File Output

test.bmp

Remove all

Obrázok 5.14 Grafické rozhranie pre prijímač

Ak nie je vybraný FILE protokol na oboch zariadeniach (aplikačná vrstva), je možné preniesť dáta nahraté zo súboru iba pomocou transportnej vrstvy. Takto sa zobrazia v textovom výstupe. Samozrejme prenášať týmto spôsobom iné ako textové súbory, nemá veľký zmysel. Navyše prijaté dáta v textovom poli sú síce overené, ale môžu v nich chýbať pakety. Preto je táto možnosť hlavne na testovanie, aby sa nemusel zadávať rozsiahly text do textového poľa.

Ak je vybraný protokol na prenos súborov iba vo vysielajúcej, prijímajúcej strane sa súbor zobrazí v textovom poli a na jeho začiatku bude JSON hlavička. Ak je naopak protokol vybraný iba v prijímajúcej, tak užívateľ nedokáže prijať žiadny UTF-8 text, keďže obslužný program očakáva hlavičku a indexovanie paketov. Tieto prípady sú chybné a pre správne fungovanie musí byť na oboch stranách nastavené to isté.

Je možné si všimnúť, že na obrázku prijímajúcej je maximálna rýchlosť oveľa väčšia ako maximálna teoretická rýchlosť prenosu. Táto chyba je spôsobená vyrovnávacou pamäťou sériového portu a jej nepravidelným a nečastým čítaním. V JavaScripte neexistujú prerušenia, všetko sa vykonáva sériovo na jednom jadre. Našťastie existuje niečo ako obsluha udalosti (z angličtiny „*event handler*“), ktorú je možné priradiť k rôznym udalostiam. Napríklad stlačenie tlačidla v prehliadači, alebo na časovač. Zjednodušene povedané, rozdiel od prerušenia je ten, že udalosť sa neobslúži, ihneď ale až po čase keď na ňu príde rad. To znamená, že ak by niekde v JS kóde bola zacyklená funkcia, celý JavaScript zastane a celé okno prehliadača zamrzne. Problém nastáva, keď príde podnet na obsluhu udalosti, ale rad je relatívne plný. Toto sa deje aj funkcii, ktorá berie dáta zo sériového portu počítača a posúva ich ďalej, aby boli prístupné z kódu prehliadača. Niekedy sa vykoná neskoro a zoberie z vyrovnávacej pamäte všetky dáta, čo sa v nej stihli nazbierať. Žiaľ prehliadač neposkytuje žiadne informácie o počte prijatých bitov do pamäte počítača a jediný spôsob, ako je možné určiť rýchlosť, je vydeliť počet nazbieraných bitov jednotkou času. Keďže spomínanú funkciu volá prehliadač s veľkým časovým rozstupom a ešte sa stáva že sa omešká, výsledná rýchlosť osciluje. S akým konkrétnym časovým rozstupom sa volá táto funkcia nie je známe. Možnosťou by bolo priemerovať rýchlosť na strane prijímajúcej za dlhšie obdobie, napríklad za 2,5 sekundy namiesto 250 ms. Celý prenos súboru trvá väčšinou kratšie ako 10 sekúnd, a teda rýchlosť by sa počas prenosu zobrazila iba párkrát. Našťastie rýchlosť meraní na strane vysielajúcej stačí priemerovať za 250 ms, pretože sa meria vo funkciách, ktorá sa volá často a to v 7 ms intervaloch. Takže pri testovaní sa bude merať rýchlosť prenosu na strane vysielajúcej.

6 TESTOVANIE

Hotový výrobok je potrebné overiť, či vyhovuje požiadavkám zo zadania a zistiť jeho limity. Skúmaná bola maximálna vzdialenosť prenosu, svetelná saturácia, rýchlosť prenosu a odolnosť voči chybám. Na konci kapitoly je tabuľka s cenou jednotlivých komponentov.

6.1 Rýchlosť prenosu

Rýchlosť prenosu sa meria automaticky v transportnej vrstve ako na strane vysielača, tak aj na strane prijímača. Žiaľ kvôli dôvodu rozobratému na konci podkapitoly Grafické rozhranie, sa dá spoliehať iba na rýchlosť vysielača. Pri prenose sa najvyššia dosiahnutá rýchlosť ukladá automaticky. Tá bola približne 9,5 kB/s, takže takmer 10-krát viac ako podmienka zo zadania. Teoretická rýchlosť (4.46) vyšla o niečo vyššie (11,4 kB/s), ale nie je ďaleko od skutočnosti. To, že skutočná rýchlosť je o niečo menšia, je spôsobené väčšou medzerou medzi paketmi. Nie je dobré ísť na doraz a riskovať, že pretečie vyrovnávací pamäť mikrokontroléru.

6.2 Dosah prenosu

Dosah prenosu závisí od kvality laseru, výkonu laseru, kvality optiky a útlmu prostredia. Toto zariadenie je určené na vnútorné použitie, preto bol dosah meraný vo vnútorných priestoroch. Prenos bol testovaný odoslaním 50 kB súboru. Optika laseru je ručne dostaviteľná a podarilo sa ju dostaviť tak, že prenos fungoval na 40 metrov. Aj keď vo vzdialenosti 40 metrov sa už lúč značne rozptyľoval (s priemerom približne 5 cm). Pre väčšie vzdialenosti by bolo vhodné vymeniť laserovú diódu.

6.3 Svetelná saturácia

Tak ako bolo vysvetlené v podkapitole Solárny panel, svetelná saturácia dokáže znemožniť prenos. Bol to kompromis medzi zvýšením prenosovej rýchlosti a odolnosti proti statickému osvetleniu. Prenos bol testovaný odosielaním textového reťazca. Ako zdroj svetla bola použitá biela výkonová LED dióda. Tá bola postupne posúvaná bližšie, až kým sa odoslaný textový reťazec nezobrazil na výstupe. Hraničná hodnota intenzity osvetlenia bola 40000 lux. Pre perspektívu, priame slnečné svetlo má približne 100000 lux a ak je zamračené tak iba 10000 lux. Vo vnútorných priestoroch (ďalej od priameho slnečného žiarenia z okien) bude zariadenie bez problémov fungovať.

6.4 Odolnosť voči chybám

V ideálnych podmienkach sa nestalo, že by nedošlo k prijatiu súboru, respektíve textového reťazca. Preto s cieľom overenia odolnosti prenosu bolo na prijímač popri prijímaní textových reťazcov svietené aj druhým laserom. Druhý laser bol vybavený s rovnakou elektronikou akú má vysielač. Odosielaný textový reťazec obsahoval iba jeden znak (prenos je najviac náchylný na chybu práve pri prenose jedného znaku). Tento znak sa odosielať každých 250 ms po dobu 10 minút (takže celkovo 2400 znakov, každý vo vlastnom pakete). Modulovaný signál z druhého laseru sa menil v náhodných časových intervaloch od 5 do 500 μ s. Algoritmy CRC, Hammingov kód a kontrolný súčet odhalili každý poškodený paket. V textovom výstupe sa po desiatich minútach vždy nachádzalo približne od 100 do 250 znakov, ktoré druhý laser nepoškodil. Parita žiaľ nedokázala ochrániť každý paket a na výstupe bolo približne 6 % znakov, ktoré neboli odoslané, ale vznikli poškodením paketu.

6.5 Cena

Posledná, ale nie menej, je cena celého zariadenia. V cene je uvedený iba materiál, z ktorého je zariadenie postavené. Nie sú v nej zahrnuté položky ako napríklad poštovné, alebo minútý spotrebný materiál (spájka, tavidlo atď...). Takže celková cena výroby bola o niečo málo vyššia.

Tabuľka 6.1 Cena zariadenia

položka	cena
laserová dióda	0,2€
solárny panel	0,07€
súčiastky modulátora	2,55€
súčiastky demodulátora	2,25€
2x SoC mikrokontrolér	7,56€
cena PLA na krabičky	1,01€
spolu	13,64€

Najväčšia položka sú SoC mikrokontroléry. Táto cena by sa dala znížiť kúpením samostatného čipu a jeho osadením priamo na dosku s ostatnou elektronikou, poprípade aj zmenou tohto čipu za lacnejší variant, no stále dostatočne rýchly. Súčiastky boli kupované za maloobchodné ceny.

ZÁVER

Táto diplomová práca popisuje výrobu zariadenia na jednosmerný prenos údajov. V práci je použitých viacero technológií, od návrhu elektroniky, cez 3D modelovanie a nízkoúrovňové programovanie, až po programovanie vo vysokoúrovňovom jazyku. Zariadenie pozostáva z vysielača a prijímača, čím je schopné jednosmerne prenášať dáta z jedného počítača na druhý. Dáta sú pri prenose zabalené v paketoch a prenášajú sa laserovým lúčom. Zdrojom laserového lúča je laserová dióda s výkonom pod 5 mW. Prijímač, na ktorý dopadá lúč, obsahuje solárny panel s rezonančným obvodom. Dáta sú modulované metódou ASK pomocou kruhového oscilátora. Najrozsiahlejšou časťou práce je návrh elektroniky modulátora a demodulátora. Túto elektroniku obsluhujú mikrokontroléry, ktoré zároveň komunikujú s počítačmi cez virtuálny sériový port. Obidva zariadenia sú napájané cez konektor USB. Obslužná aplikácia je napísaná v jedinom HTML dokumente, ktorý sa otvorí v prehliadači. Po spárovaní s prijímačom, alebo vysielačom sa stránka odpovedajúco nakonfiguruje. V obslužnej aplikácii je možné si zvoliť jeden z algoritmov. Na výber je parita, kontrolný súčet, Hammingov kód a cyklický redundantný súčet. Aplikácia v prehliadači sa stará iba o prenos dát. Generovanie redundancie a overovanie paketov sa odohráva v nižšej vrstve implementovanej v mikrokontroléroch. Prenášať je možné UTF-8 reťazce a súbory. Na strane vysielača aj prijímača sa samozrejme musí vybrať rovnaký algoritmus. Zmeraná maximálna rýchlosť prenosu je 9,5 kB/s a maximálna vzdialenosť prenosu je 40 metrov. Týmto sú splnené požiadavky zo zadania. Celková cena za použitý materiál bola 13,64 €. Ak by boli súčiastky kupované za veľkoobchodné ceny a mikrokontroléry by boli na doske spolu s ostatnou elektronikou, cena by bola aspoň o tretinu nižšia.

Prenos je takmer vždy zaťažený šumom. Preto je potrebné zvoliť overovací algoritmus, ktorý chyby zaručene odhalí. Pri meraní všetky algoritmy okrem parity dokázali dáta ochrániť. Avšak pre toto zariadenie je najviac vhodný CRC algoritmus. Okrem toho, že neberie výpočtový čas procesoru, keďže má mikrokontrolér na jeho počítanie vlastné hardvérové vybavenie, má najvyššiu Hammingovu vzdialenosť a teda poskytuje najvyššiu ochranu. Dobrá zmena by bola vymeniť mikrokontrolér za iný, v ktorom je možné si nastaviť vlastný deliaci polynóm.

Téma tejto diplomovej práce je zaujímavá z toho pohľadu, že solárny panel je lacná súčiastka, ktorú ako bolo dokázané, je možné využiť nie len na získavanie energie,

ale aj na prenos dát. Vložiť solárny panel do rezonančného obvodu, prináša nový spôsob jeho využitia. V budúcnosti bude mať veľa domácností solárne panely na strechách domov, popríklad bytových objektov. S vhodným návrhom panelov, tak aby ich parazitná kapacita bola čo najmenšia, by ich bolo možné využiť aj na prenos dát. Jediná potrebná vec by bol dátový modul, ktorý by sa zapojil medzi solárny panel a regulátor nabíjania. Takto sa zo strechy v podstate stane jeden veľký prijímač. Iný druh použitia môže byť v priestoroch, v ktorých sa ťažko šíri rádiové vlnenie.

LITERATÚRA

- [1] Základy fotodiód. *WAVELENGTH ELECTRONICS* [online]. Bozeman, Montana: Wavelength Electronics, 2008 [cit. 2020-12-18]. Dostupné z: <https://www.teamwavelength.com/photodiode-basics/>
- [2] MARATHE, Niteesh. Extrinzické polovodiče. Engineering LibreTexts [online]. University of California: LibreTexts libraries, 2020 [cit. 2020-12-18]. Dostupné z: [https://eng.libretexts.org/Bookshelves/Materials_Science/Supplemental_Modules_\(Materials_Science\)/Semiconductors/Extrinsic_Semiconductors](https://eng.libretexts.org/Bookshelves/Materials_Science/Supplemental_Modules_(Materials_Science)/Semiconductors/Extrinsic_Semiconductors)
- [3] A Primer on Photodiode Technology. San Diego Education [online]. San Diego: University of San Diego, 2002 [cit. 2020-12-18]. Dostupné z: <http://home.sandiego.edu/~ekim/photodiode/pdtech.html>
- [4] How Solar Panels Works. *AUTODESK* [online]. California, U.S.: Autodesk, 2016 [cit. 2020-12-18]. Dostupné z: <https://www.autodesk.com/products/eagle/blog/solar-panels-solar-electricity-works/>
- [5] Electronic Devices: pn junction capacitance - Depletion and Diffusion. In: *YouTube* [online]. India: YouTube, 2015 [cit. 2020-12-18]. Dostupné z: <https://www.youtube.com/watch?v=v1pnwGgCyPU>
- [6] Response of a silicon photo diode vs wavelength of the incident light. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2009, 25 April 2009 [cit. 2020-12-18]. Dostupné z: https://en.wikipedia.org/wiki/Photodiode#/media/File:Response_silicon_photodiode.svg
- [7] *Diode lasers and photonic integrated circuits*. 2nd ed. Hoboken: Wiley, 2012. Series in microwave and optical engineering (Wiley). ISBN 978-0-470-48412-8.
- [8] Photodiode Characteristics and Applications. *OSI Optoelectronics* [online]. California, U.S.: OSI Optoelectronics, 2006, 2006 [cit. 2020-12-18]. Dostupné z: <http://www.osioptoelectronics.com/application-notes/an-photodiode-parameters-characteristics.pdf>
- [9] FLOYD, Thomas L. *Electronic devices: conventional current version*. Tenth edition. Harlow, England: Pearson, [2018]. ISBN 978-129-2222-998.
- [10] ATKIN, Justin. Receiving Data from Satellites. *Thought Emporium* [online]. Toronto, Ontario, Canada: Thought Emporium, 2017 [cit. 2020-12-18]. Dostupné z: <https://www.thethoughtemporium.com/radio>

- [11] Sériový přenos dat. *TECHNICKÁ UNIVERZITA OSTRAVA* [online]. Ostrava: Katedra informatiky, 2005 [cit. 2020-12-18]. Dostupné z: <http://www.cs.vsb.cz/grygarek/LAN/sem/sercomm.html>
- [12] Resonance. *GETMYUNI* [online]. India: GETMYUNI, 2018 [cit. 2020-12-18]. Dostupné z: https://getmyuni.azureedge.net/assets/main/study-material/notes/electrical-engineering_engineering_network-analysis_resonance_notes.pdf
- [13] SHARMA, R. K., D. R. BHASKAR, V. K. SINGH a Raj SENANI. *Sinusoidal Oscillators and Waveform Generators Using Modern Electronic Circuit Building Blocks*. New York City: Springer, November 2015. ISBN 9783319237121.
- [14] FÖLL, H. Laser Diodes: Stimulated Emission and Inversion. *Christian-Albrechts-Universität zu Kiel: Faculty of Engineering* [online]. Kiel, Germany: Christian-Albrechts-Universität zu Kiel, 2005 [cit. 2020-12-18]. Dostupné z: https://www.tf.uni-kiel.de/matwis/amat/semitech_en/kap_9/backbone/r9_2_2.html
- [15] ŠEBESTA, Vladimír. *Teorie sdělování: skripta*. Brno, 1998.
- [16] Lecture 12. In: *Eng.lbl.gov* [online]. [cit. 2020-12-18]. Dostupné z: http://www-eng.lbl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/FO/Lect12-photodiode%20detectors.pdf
- [17] Rohovkový reflex. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2016, 17. 4. 2016 [cit. 2021-5-3]. Dostupné z: https://cs.wikipedia.org/wiki/Rohovkov%C3%BD_reflex
- [18] Laser safety. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2021, 22 April 2021 [cit. 2021-5-3]. Dostupné z: https://en.wikipedia.org/wiki/Laser_safety
- [19] Quantum efficiency. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2021, 6 January 2021 [cit. 2021-5-3]. Dostupné z: https://en.wikipedia.org/wiki/Quantum_efficiency
- [20] TEXAS INSTRUMENTS INCORPORATED. SN74HC00 Technický List. 2nd ed. Dallas, Texas, 1982, 15 s. Dostupné tiež z: <https://datasheet.octopart.com/SN74HC00N-Texas-Instruments-datasheet-15794.pdf>
- [21] JANÁSEK, Vojtěch. Design of ultra low noise amplifiers. JanasCard [online]. Praha, 2012, s. 11 [cit. 2021-5-3]. Dostupné z: <http://www.janascard.cz/PDF/Design%20of%20ultra%20low%20noise%20amplifiers.pdf>

- [22] SCHRUM, Jacob. Error Detection and Correction 1: Internet Checksum. YouTube [online]. Georgetown, Texas: YouTube, 2016, 1. 6. 2016 [cit. 2021-5-3]. Dostupné z: <https://www.youtube.com/watch?v=EmUuFRMJbss>
- [23] SANDERSON, Grant. Hamming codes part 2, the elegance of it all. YouTube [online]. USA, 2020, 4. 9. 2020 [cit. 2021-5-3]. Dostupné z: https://www.youtube.com/watch?v=b3NxrZOu_CE
- [24] MatterHackers [online]. Valencia Circle, California, USA, 2013, 28. 2. 2013 [cit. 2021-5-3]. Dostupné z: <https://www.matterhackers.com/articles/how-to-succeed-when-printing-in-pla>
- [25] Hamming code. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2008, 29. 4. 2008 [cit. 2021-5-3]. Dostupné z: https://en.wikipedia.org/wiki/Hamming_code
- [26] FREED, Ned. Media Type Specifications and Registration Procedures: Naming Requirements. Internet Engineering Task Force (IETF) [online]. Middletown, NJ, USA: AT&T Laboratories, 2013 [cit. 2021-5-3]. Dostupné z: <https://tools.ietf.org/html/rfc6838#section-4.2>
- [27] Media Type Specifications and Registration Procedures [online]. Middletown, NJ, USA: AT&T Laboratories, 2013 [cit. 2021-5-3]. Dostupné z: <https://tools.ietf.org/html/rfc6838>
- [28] Media Types [online]. 2021 [cit. 2021-5-3]. Dostupné z: <https://www.iana.org/assignments/media-types/media-types.xhtml>
- [29] Internet Engineering Task Force. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2021, 1986 [cit. 2021-5-3]. Dostupné z: https://cs.wikipedia.org/wiki/Internet_Engineering_Task_Force
- [30] Cyclic redundancy check. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-5-3]. Dostupné z: https://en.wikipedia.org/wiki/Cyclic_redundancy_check
- [31] CORRIGAN, Michael a Joe TRIGGS. CRC Testing in Video Applications. Analog Devices [online]. [cit. 2021-5-3]. Dostupné z: <https://www.analog.com/en/technical-articles/crc-testing-in-video-applications.html>
- [32] RM0383 Reference manual: STM32F411xC/E advanced ARM®-based 32-bit MCUs. Dostupné z: https://www.espruino.com/datasheets/STM32F411xE_ref.pdf

- [33] EATER, Ben. How do CRCs work? YouTube [online]. 2019, 28. 4. 2019 [cit. 2021-5-3]. Dostupné z: <https://www.youtube.com/watch?v=izG7qT0EpBw>
- [34] EATER, Ben. Hardware build: CRC calculation. YouTube [online]. 2019, 2. 6. 2019 [cit. 2021-5-3]. Dostupné z: <https://www.youtube.com/watch?v=sNkERQIK8j8>
- [35] Field Axioms. Wolfram Math World [online]. [cit. 2021-5-3]. Dostupné z: <https://mathworld.wolfram.com/FieldAxioms.html>
- [36] AN4187 Application note: Using the CRC peripheral in the STM32 family. 16 s. Dostupné z: https://www.st.com/resource/en/application_note/dm00068118-using-the-crc-peripheral-in-the-stm32-family-stmicroelectronics.pdf

ZOZNAM OBRÁZKOV

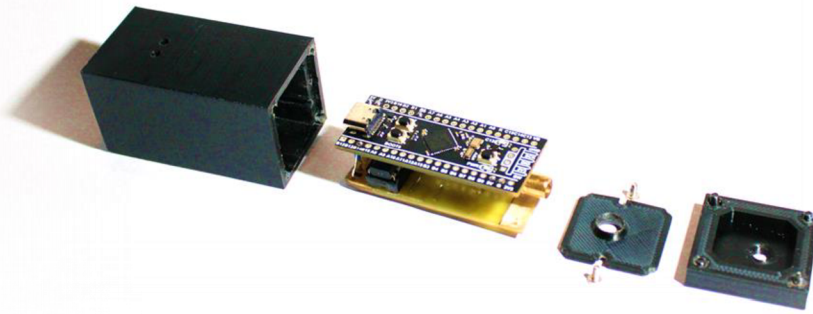
Obrázok 1.1 Prierez štruktúrou solárneho panelu [1].....	3
Obrázok 1.2 Senzitivita kremíkovej fotodiódy [6].....	5
Obrázok 1.3 Dva rôzne faktory kvality paralelného rezonančného obvodu.....	9
Obrázok 1.4 Zapojenie BJT so spoločným emitorom	10
Obrázok 2.1 Príklad platných správ overených paritou.....	13
Obrázok 2.2 Hardvérová kalkulačka parity	14
Obrázok 2.3 Ručný výpočet CRC-8	22
Obrázok 2.4 Hardvérová kalkulačka na CRC-8, 0x107	25
Obrázok 3.1 Metóda kódovania ASK.....	27
Obrázok 3.2 Bloková schéma celého systému.....	28
Obrázok 3.3 Schéma paketu pre paritu	29
Obrázok 3.4 Schéma paketu pre ostatné algoritmy	29
Obrázok 3.5 Nesynchronizované vzorkovanie	30
Obrázok 3.6 Časový priebeh prenosu	31
Obrázok 3.7 Protokol na prenos súborov.....	34
Obrázok 3.8 Súbor analyzovaný prehliadačom	36
Obrázok 4.1 Modulátor	39
Obrázok 4.2 Minimálna nastaviteľná frekvencia kruhového oscilátoru.....	40
Obrázok 4.3 Solárny panel pred a po rezaní	41
Obrázok 4.4 Náhradná schéma solárneho panelu	42
Obrázok 4.5 Kmitanie pridanej cievky a parazitnej kapacity pri nulovom predpätí	43
Obrázok 4.6 Prvý stupeň zosilňovača.....	44
Obrázok 4.7 Derivačný člen, zapojenie na zmeranie rýchlosti nábehu prúdu.....	47
Obrázok 4.8 Prúd solárneho panelu vyvolaný jednotkovým skokom laserového lúča ..	47
Obrázok 4.9 Zjednodušená schéma rezonančného obvodu so záťažou.....	48
Obrázok 4.10 Zosilnený signál dvojstupňovým tranzistorovým zosilňovačom.....	51
Obrázok 4.11 Druhá časť demodulátora.....	54
Obrázok 4.12 Tvar signálu pred bufferom.....	55
Obrázok 4.13 Konštrukcia prijímača	57
Obrázok 4.14 Konštrukcia vysielajúceho	58
Obrázok 5.1 Input Capture Mode	60
Obrázok 5.2 Vývojový diagram signálovej vrstvy prijímača	61

Obrázok 5.3 Debuggovanie pomocou osciloskopu	62
Obrázok 5.4 Vývojový diagram signálovej vrstvy vysielča	63
Obrázok 5.5 Vývojový diagram transportnej vrstvy vysielča.....	66
Obrázok 5.6 Medzera medzi paketmi	67
Obrázok 5.7 Vývojový diagram transportnej vrstvy prijímača	67
Obrázok 5.8 Vývojový diagram aplikačnej vrstvy vysielča	68
Obrázok 5.9 Vývojový diagram aplikačnej vrstvy prijímača	69
Obrázok 5.10 Prenos súboru.....	70
Obrázok 5.11 Úvodná stránka	70
Obrázok 5.12 Výber protokolu	71
Obrázok 5.13 Grafické rozhranie pre vysielča	71
Obrázok 5.14 Grafické rozhranie pre prijímač	72

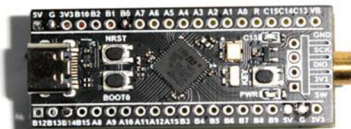
ZOZNAM TABULIEK

Tabuľka 2.1 Index s 1 na 4. mieste.....	17
Tabuľka 2.2 Index s 1 na 2. mieste.....	17
Tabuľka 2.3 Index s 1 na 3. mieste.....	17
Tabuľka 2.4 Index s 1 na 1. mieste.....	17
Tabuľka 2.5 Algebraické axiómy pre množiny [35].....	21
Tabuľka 3.1 Textové reťazce pre jednotlivé protokoly	32
Tabuľka 3.2 Niektoré MIME typy	35
Tabuľka 3.3 Vrstvy zariadenia	37
Tabuľka 4.1 Parametre 3D tlače	58
Tabuľka 5.1 Parametre pre CRC-32/MPEG-2.....	65
Tabuľka 6.1 Cena zariadenia	75

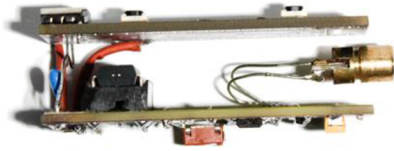
PRÍLOHY



Príloha 1 Konštrukcia vysielača



Príloha 2 Elektronika vysielača, náhľad zhora



Priloha 3 Elektronika vysieláča, náhľad z boku



Priloha 4 Elektronika vysieláča, náhľad zospodu



Príloha 5 Zložený vysielateľ, 3D náhľad



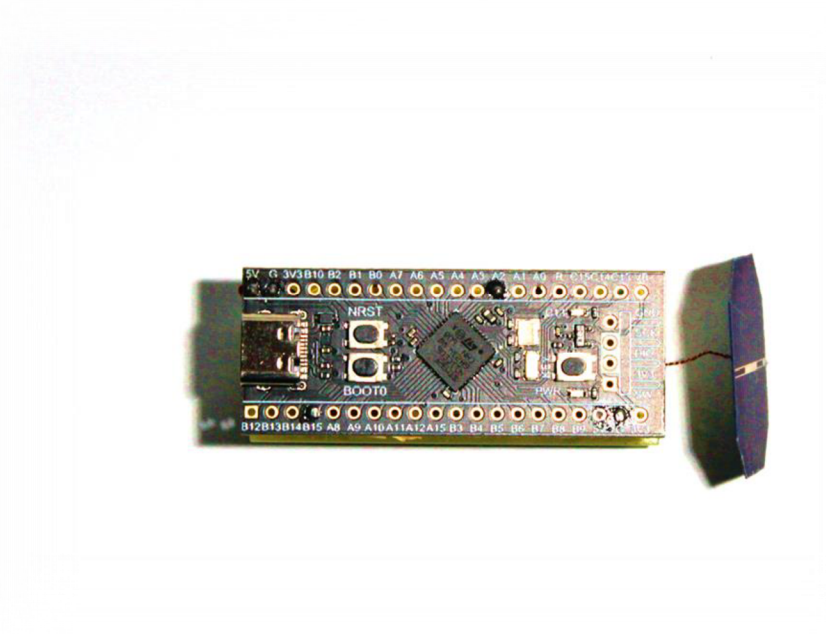
Príloha 6 Zložený vysielateľ, náhľad zhora



Príloha 7 Zložený vysielač, náhľad zozadu



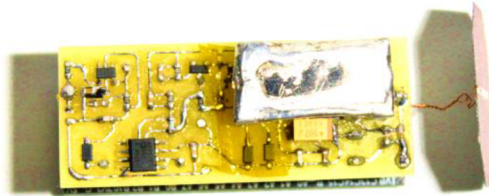
Príloha 8 Konštrukcia prijímača



Príloha 9 Elektronika prijímača, náhľad zhora



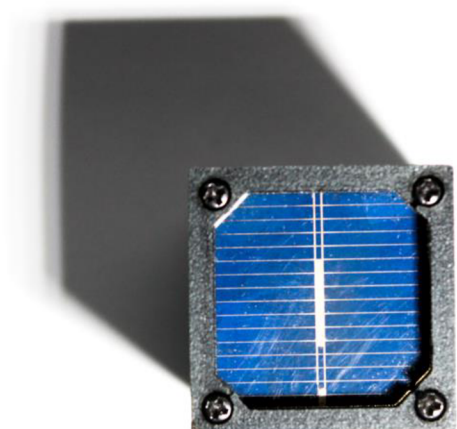
Príloha 10 Elektronika prijímača, náhľad z boku



Príloha 11 Elektronika prijímača, náhľad zospodu



Príloha 12 Zložený prijímač, 3D náhľad



Príloha 13 Zložený prijímač, náhľad zhora



Príloha 14 Zložený prijímač, náhľad zozadu