

Duplexný prenos audiosignálov ethernetom

Olexa Jakub · Elektrotechnika, Študentské práce

14.06.2010



So zväčšujúcou sa výpočtovou rýchlosťou, možnosťami prepojenia počítačov a so zvyšujúcou sa cenovou dostupnosťou sa tiež zväčšuje aj záujem širokej verejnosti a obyčajných užívateľov o vzájomnú komunikáciu na veľké vzdialenosti. Najmodernejšiu a čím ďalej populárnejšiu formu komunikácie dvoch alebo viacerých užívateľov dnes predstavuje internet, a to práve pre jeho efektívnosť, vysokú rýchlosť, možnosť prenosu veľkého objemu dát, nízku cenu, prístup a dostupnosť k hociktorému internetovému užívateľovi na svete, a najmä pre rozširujúcu sa dostupnosť ethernetových sieťových rozhraní, a to nie len v domácnostiach, ale aj iných technických odvetviach, ako napríklad riadiace a komunikačné ethernetové zbernice vo vlakových súpravách.

1. Úvod

Rýchle sieťové rozhrania a prenosové vedenia, či už koaxiálne, alebo optické dnes umožňujú dátový prenos až rádovo Gbit/s a stávajú sa základom počítačovej a sieťovej integrácie. Internet a ethernetové siete dnes z oblasti audio a video aplikácií vďaka svojmu rozšíreniu umožňujú užívateľom medzinárodné vysielanie rádiových staníc, streamovanie, nahrávanie a zdieľanie hudby, videí, filmov, alebo aj telefonovanie. Najmä telefonovanie cez internet alebo aj inak nazývaná "voice over IP" (skr.: VOIP) sa stáva čoraz populárnejším najmä pre jej nízku prevádzkovú cenu, široký sortiment služieb, ale aj pre veľký sortiment rôznych aplikácií a zariadení podporujúcich VOIP. Tento sa už pomaly stáva štandardom v rámci ethernetových LAN sietí menších, ale aj väčších firiem, ktoré používajú práve VOIP telefóny pre vnútrofirminú komunikáciu.

Cieľom tejto práce je demonštrovať možnosti prenosu zvuku cez ethernetovú zbernicu a spôsoby spracovania nízkofrekvenčných audiosignálov vo zvolenej oblasti hovorového pásma 300Hz až 4000Hz, čo je šírka frekvenčného pásma dostačujúca na zrozumiteľnú komunikáciu medzi účastníkmi pri relatívne nízkom dátovom toku. Pri návrhu bola použitá vývojová doska s mikropočítačom Atmega32 taktovaného na 16MHz a periférny ethernetový modul so sieťovým radičom ENC28J60 s možnosťou prenosu 10Mb/s (10BASE-T). Tieto boli vyvinuté na Katedre rádioelektroniky a zapožičané pre realizáciu tejto práce.

Komunikačný program bol navrhnutý pre komunikáciu medzi modulom a počítačom a bol naprogramovaný vo vývojovom prostredí AVR studio 4 s kompilátorom AVR/GCC v jazyku C. Ako základ pre túto prácu slúžil program pre riadenie a monitorovanie

prostredníctvom LAN sietí, ktorý využíva knižnice vyvinuté firmou Microchip pre daný sieťový radič [7].

2. Ethernet a TCP/IP protokol

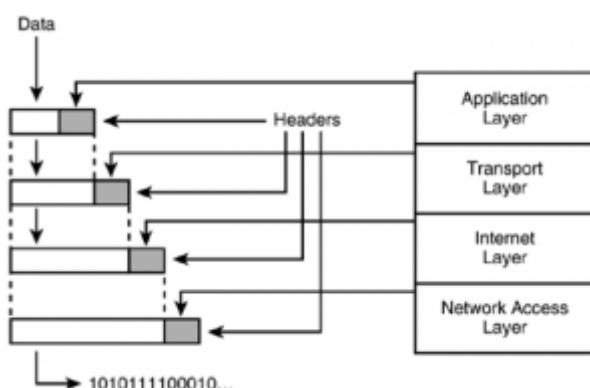
Ethernet je základnou normou definujúcou pripájanie viacerých zariadení na spoločnú ethernetovú zbernicu za účelom vytvorenia siete. Je štandardizovaný ako IEEE 802.3 pre lokálne siete typu LAN. Definuje topológiu, rozhrania, komunikačný protokol sietí, ako aj napäťové úrovne signálov alebo použité konektory. LAN siete slúžili pôvodne na prepájanie menšieho počtu počítačov na formovanie lokálnej siete, ale s rozšírením internetu sa tieto siete prepájajú s ostatnými sieťami WAN pomocou routerov (gateway). Router je zariadenie slúžiace na prekladanie protokolov a pripájanie LAN k väčším sieťam. Tak je zároveň zabezpečená komplexná podpora protokolu TCP/IP.

Protokol TCP/IP je odvodený z OSI modelu. Jeho základnými funkciami a vlastnosťami je najmä decentralizované riadenie siete, logické adresovanie, odhaľovanie chýb prenosu, riadenie toku dát a potvrdzovanie prijatých správ, ako aj ARP (address name resolution), čiže mapovanie IP adresy pomocou fyzickej MAC adresy. Táto funkcia umožňuje urýchliť vyhľadanie jednotlivých zariadení v sieti.

TCP/IP je hierarchický 4 vrstvový protokol. Jeho jednotlivé vrstvy využívajú služby subvrstiev, a zároveň ich poskytujú nadradeným vrstvám. Každá vrstva k posielaným dátam pridáva svoju vlastnú hlavičku s potrebnými údajmi na zabezpečenie bezpečnej, bezporuchovej komunikácie s korešpondujúcou vrstvou, ktorej je správa určená. Užitočné dáta sa takto posúvajú dolu stackom až sú nakoniec poslané na fyzickú zbernicu a spätne dekódované súvisiacimi vrstvami hierarchicky zdola nahor na prijímacej strane. Takýto vrstvový systém umožňuje univerzálnosť a jednoduchosť začlenenia protokolu do jednotlivých aplikácií. [1]

2.1. Vrstvy protokolu TCP/IP

Jednotlivé vrstvy protokolu a ich hierarchia je zobrazená na obr. č.1 od najnižšej (fyzickú) až po najvyššiu (aplikačnú) vrstvu.



Obr. 1. TCP/IP vrstvový protokol [1]

Fyzická vrstva je softvérovo/hardvérová a zabezpečuje fyzické pripojenie zariadenia na zbernicu a riadenie funkcií sieťového adaptéra. Úlohou tejto vrstvy je formátovať a rozdeľovať posiellané dáta na rámce a tieto konvertovať do sledu bitov posiellaných na zbernicu. Dáta sú prenášané asynchrónne po dvoch symetrických pároch: pre

vysielanie Rx a prijímanie rámcov Tx. Taktiež zaobstaráva preposielanie chybných, ako aj základnú kontrolu prijímaných rámcov. Hlavička pridaná fyzickou vrstvou obsahuje okrem iných informácií aj unikátnu 48 bitovú adresu sieťového radiča (MAC adresu) oboch strán komunikácie, ktorá je dôležitá pri identifikácii recipienta správy a komunikáciu v rámci lokálnej siete.

Internetová vrstva zabezpečuje komunikáciu cez routre mimo lokálnu sieť, subnet. Na routovanie datagramov sa už nepoužíva MAC adresa ale 32 bitová IP adresa, pridelená pre každý počítač v sieti a 32 bitová subnet maska charakterizujúca lokálnu sieť. Špeciálny typ IP adresy pozostáva z jednotkových binárnych oktetov a používa sa na vysielanie datagramov pre všetkých účastníkov siete. Internetová vrstva poskytuje funkcie ICMP, ARP a RARP protokolov. Pri použití ARP protokolu sa MAC adresa stáva pre užívateľa vyšších vrstiev transparentná.

Transportná vrstva slúži na vytvorenie, udržanie a ukončenie komunikácie s jedným alebo viacerými zariadeniami na sieti. Jej úlohou je multiplexovanie/demultiplexovanie správ z viacerých aplikácií, a tiež kontrola chýb. V hlavičke datagramu sú obsiahnuté informácie o porte vysielacej a prijímacej aplikácie. Port je predefinovaná adresa aplikácie a slúži ako most medzi transportnou a aplikačnou vrstvou. Základné protokoly, ktoré táto vrstva poskytuje sú UDP a TCP protokoly. Pre túto prácu bolo zvolené použitie UDP protokolu transportnej vrstvy, lebo nieje spojovo orientovaný v porovnaní s TCP, nepotrebuje nadväzovať spojenie a ani ho udržiavať, čo zvyšuje rýchlosť prenosu. Na druhej strane UDP protokol poskytuje len základnú kontrolu chýb matematickým výpočtom checksumu. Dodatočná kontrola môže byť pridaná, ale na úkor veľkosti prenášanej užitočnej informácie v datagrame . [1]

Aplikačná vrstva je najvrchnejšou vrstvou TCP/IP protokolu. Úlohou samotnej aplikácie je vykonávať konkrétne stanovené úlohy nad ostatnými vrstvami, a tak zaobstarat priame rozhranie pre užívateľa, ako aj poskytovať vlastné funkcie iným aplikáciám [1].

3. AD a DA prevodníky

Jednou z dôležitých schopností každého zariadenia spracovávajúceho audio signály je možnosť vzorkovať a kvantovať spojitý analógový signál, a tak ho transformovať do diskretného číslicového signálu. Jednotlivé hodnoty sú zakódované do dvojkovej sústavy a ďalej je možné s nimi narábať ako s binárnymi číslami už v rámci číslicového systému. Túto funkciu umožňujú AD prevodníky. Existujú viaceré možnosti AD prevodu. Jednou z možností môže byť kompenzačný AD prevodník s postupnou aproximáciou, ktorý často býva integrovanou súčasťou mikropočítačových systémov. V tejto páci bol využitý práve tento typ integrovaného interného prevodníka mikropočítača Atmega32. Kompenzačné AD prevodníky sú dostatočne rýchle a pri použití stabilizovanej napäťovej referencie aj presné. Preto môžu byť použité aj pre prevod audio signálov.

Súčasťou digitálnych audio-systémov býva aj DA prevodník, ktorého úlohou je transformovať diskretné číslicové signály na spojitý analógový signály a tak umožniť vybudenie sluchového vnemu pomocou elektroakustického meniča. Najjednoduchším typom DA prevodníka je paralelný prevodník zostrojený odporovou sieťou z oporov $2R$ -

R. Takéto prevodníky sú významné svojou jednoduchou konštrukciou a veľkou rýchlosťou prevodu, ale na druhej strane zase aj nelinearitami, tepelným šumom a odchýlkami hodnôt rezistorov. Keďže rada mikropočítačov Atmega 32 nemá implementovaný takýto typ prevodníka, bolo nutné ho implementovať do periférneho modulu. Na účel spracovávanía audiosignálov sa v tejto dobe skôr využívajú sigma/delta a delta/sigma prevodníky a to najmä pre ich vysoký odstup signál/šum a dobré dynamické vlastnosti [2].

3.1. Vzorkovanie a aliasing

Vzorkovanie je proces, pri ktorom sa zo spojitého signálu v čase stáva diskretný. Pri vzorkovaní dochádza k násobeniu analógového signálu s postupnosťou impulzov o určitej frekvencii. Podľa Nyquistovej teórie je táto vzorkovacia frekvencia zdola ohraničená a to hodnotou dvakrát väčšou než je frekvencia analógového signálu, ktorý AD prevodník vzorkuje. Vo frekvenčnej oblasti sa po navzorkovaní vytvoria identické obrazy v okolí vzorkovacej frekvencie a jej násobkom.

Vzorkovanie pri komparačných prevodníkoch je najčastejšie realizované s pomocou S/H (sample and hold) obvodu na vstupe AD prevodníka, ktorý zaručuje konštantnú hladinu meranej analógovej veličiny počas prevodu. Na navzorkovanie signálu stačí spúšťať AD prevod periodicky podľa vzorkovacej frekvencie. Pri vzorkovaní analógového signálu musí byť definovaná šírka frekvenčného pásma.

Ak totiž horná frekvencia analógového signálu bude väčšia ako frekvencia definovaná podľa Nyquistovho kritéria, dôjde k prekrytiu frekvenčných spektier, a teda k aliasingu. Aliasingu sa dá zabrániť vhodným vstupným filtrom, ktorý utlmí frekvencie nad Nyquistovu frekvenciu. Taktiež je možné zväčšiť vzorkovaciu frekvenciu (takzvaný oversampling) pre zníženie nárokov na antialiasingové filtre. Pri oversamplingu sa zároveň zlepšujú šumové vlastnosti, keďže sa konštantne rozložený biely šum rozloží do väčšieho počtu frekvenčných obrazov, ktoré sú potom odfiltrované. [3]

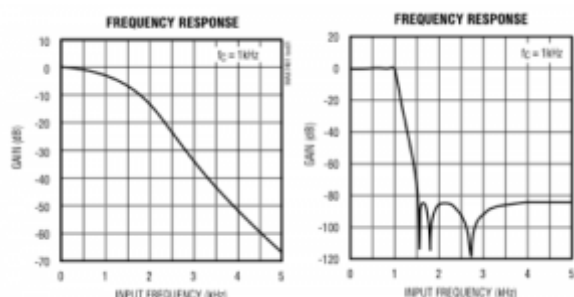
3.2. Antialiasingové a interpolačné filtre

Antialiasingový filter je filter typu dolná priepusť a jeho medzná frekvencia je menšia ako Nyquistova frekvencia. V oblasti nad Nyquistovu frekvenciu by sa malo nachádzať pásmo útlmu filtra. Keďže zvyčajne majú audiosignály veľký odstup signál/šum (80dB a viac) je nutné použiť filter s úzkym prechodovým pásmom. Jednou z možností je použiť filtre so strmou modulovou frekvenčnou charakteristikou s čo najmenším zvlnením v oblasti prepúšťania a čo najmenšie skupinové oneskorenie. Najčastejšie sa používajú frekvenčné charakteristiky Beselovej a eliptickej aproximácii filtra.

Obsahom frekvenčného spektra výstupu z AD prevodníka je ako spektrum pôvodného analógového signálu, tak aj jednotlivé frekvenčné obrazy okolo vzorkovacej frekvencie a jej násobkom. Napriek tomu, že sa tieto komponenty môžu nachádzať mimo frekvenčné pásmo audio signálov, a teda nie sú počuteľné, je vhodné ich zo signálu kvôli lepším šumovým pomerom a kvôli možným následným zosilňovacím stupňom odstrániť. V prípade, že časť frekvenčného obrazu zasahuje do počuteľnej oblasti spektra je použitie dolnopriepustného interpolačného filtra nutnosťou.

Na filtráciu sa používajú filtre s čo najstrmšou modulovou frekvenčnou

charakteristikou. Plochá fázová charakteristika a malé zvlnenie v pásme prepúšťania sú kritické pre lepšie zrekonštruovanie vyšších frekvenčných komponentov. Ak je v oblasti medznej frekvencie výrazné fázové zvlnenie, výsledný zvuk má podľa audiofilov “plechový” charakter. Preto výhodnejšie z tohto hľadiska sú Besselove alebo Butterworthove filtre vyšších rádo. [3]



Obr. 2. Frekvenčné charakteristiky (v poradí zprava) eliptického (MAX7400) a Besselovho filtra obvodov (MAX7401) [6]

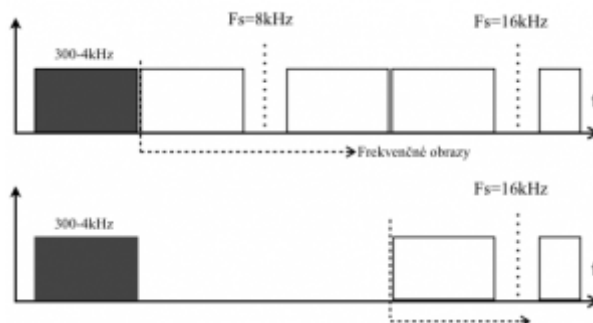
4. Návrh AD/DA modulu

Základnou možnosťou vývojovej dosky pre komunikovanie s inými zariadeniami prostredníctvom audio datagramov by mala byť možnosť načítať, prehrať a spätne zrekonštruovať vzorky analógových audio signálov. Na tento účel som navrhol modul dvoch oddelených vstupno/výstupných častí pre základnú úpravu vstupných a výstupných audiosignálov spolu s riadiacim programom. Modul je schopný spracovávať a rekonštruovať audio signály v oblasti 300Hz až 4kHz, čo predstavuje znížené nároky pre vzorkovaciu frekvenciu a jednotlivé filtre, rýchlosť spracovávania vzoriek mikropočítačom ako aj zníženú veľkosť dátového toku. Modul umožňuje použiť ako zdroj signálu elektretový mikrofón alebo aj výstup zo zvukovej karty počítača. K výstupnému obvodu je možné pripojiť cez výstupný transformátor reproduktor alebo slúchadlá, alebo je možné pripojiť externý zosilňovač.

4.1. Realizácia AD časti modulu

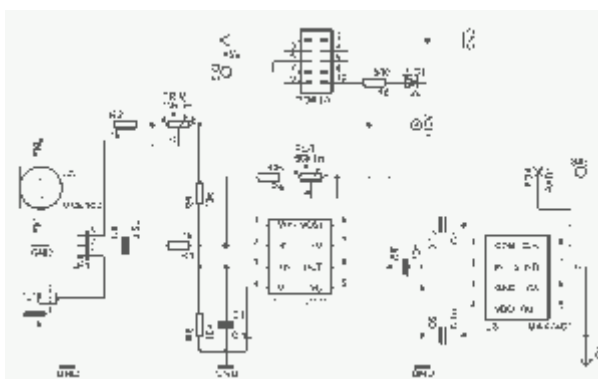
Pre jednoduchšie spracovávanie a prehrávanie vzoriek pri relatívne nízkom kvantizačnom šume som zvolil rozlíšenie 8 z možných 10 bitov interného AD prevodníka v Atmega32, čo umožňuje aj použitie vyšších vzorkovacích frekvencií ako uvedených 15kHz v popise mikropočítača. Vzorkovanie vstupných signálov je realizované periodickým spúšťaním AD prevodníka. Na tento účel je použitý časovač Timer/Counter0 ako interný zdroj prerušenia s vektorom SIG_OUTPUT_COMPARE0.

Keďže maximálna akceptovateľná načítavaná frekvencia spektra je 4kHz, potom vzorkovacia frekvencia podľa Nyquistovej teórie musí byť aspoň 8kHz. Takýto návrh by vyžadoval veľmi strmé charakteristiky antialiasingového a interpolačného filtra a preto bola vzorkovacia frekvencia nastavená na 16kHz (Obr. 3).



Obr. 3. Frekvenčné spektrum pri vzorkovacej frekvencií 8 kHz (hore) a 16kHz (dole)

Za referenčný zdroj pre konverziu je zvolené 5-voltové stabilizované a filtrované napätie vývojovej dosky. Vzhľadom na to, že je využitých 8 bitov z celkového rozsahu prevodníka bude môcť vstupný signál nadobúdať hodnoty 0-4V a bude ho nutné vystrediť. S hodnotou výsledného registra sa potom môže narábať vo vektore prerušenia SIG_ADC. Vstupný signál sa pripája na PORTA PA3.



Obr. 4. Schéma zapojenia AD časti modulu

Podľa schémy na Obr. 4. je možné pomocou JP1 jumperu zvoliť zdroj vstupného signálu. Elektretový mikrofón je napájaný 10V napätím cez odpor R2 a trimrom TRIM, ktorý slúži na reguláciu citlivosti. Amplitúda výstupného signálu mikrofónu zameraná pri maximálnej citlivosti je približne 50mV. Pri nízko-frekvenčných aplikáciách je kvôli odstráneniu skreslenia, a tiež "plechového" zvuku pri vyšších frekvenciách, vhodné použiť rýchle operačné zosilňovače.

Na zosilnenie signálu bol preto použitý nízkošumový rýchly operačný zosilňovač LT1007 od firmy Linear Technology v invertujúcom zapojení. Toto zapojenie bolo doplnené o napäťový delič s blokovacím kondenzátorom, ktorý privádza polovičnú hodnotu vstupného napätia na neinvertujúci vstup operačného zosilňovača. Takéto zapojenie umožňuje použiť iba jeden napájací zdroj pripojený na vstup kladného napájacieho napätia zosilňovača. Vstup pre záporné napájacie napätie je potom pripojený na zem.

Obvod LT1007 môže pracovať s napájacími napätiami do $\pm 22V$ a má lepšie šumové vlastnosti pri vyšších napájacích napätíach ako 5V. Keďže napájacie napätie vývojovej dosky je práve 5V, bolo zdvojnásobené pomocou obvodu LTC660 od firmy Linear Technology, a týmto napájať operačný zosilňovač a mikrofón (nie je súčasťou Obr. 4.). Zosilnenie invertujúceho operačného zosilňovača je navrhnuté tak, aby sa dalo meniť pomocou lineárneho potenciometru POT. Vstupný signál je tak možné zosilniť 40 až 100-krát. Na odfiltrovanie frekvenčných komponentov pod 300Hz, a tiež ako kapacitná

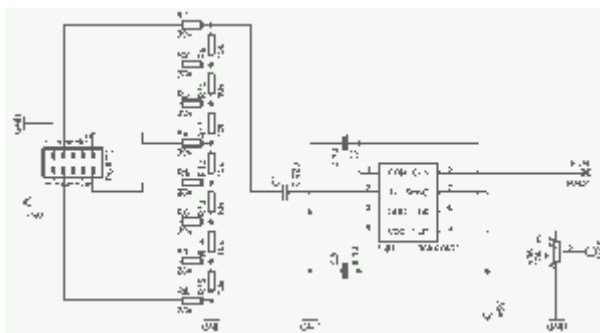
väzba na odstránenie jednosmernej zložky, slúži kondenzátor C4 tvoriaci hornopriepustný filter 1.rádu.

Filtráciu vyšších frekvenčných komponentov audiosignálov a antialiasing zabezpečuje obvod MAX7400[6] s eliptickou frekvenčnou charakteristikou od firmy Maxim. Tento filter je typu dolná priepusť so spínanou kapacitou. Medzná frekvencia filtra sa nastavuje pomocou externého hodinového signálu z vývojovej dosky v rozmedzí 1kHz až 5kHz. Keďže filter obsahuje vnútornú frekvenčnú deličku 1:100 musí byť frekvencia 100-krát vyššia ako požadovaná medzná frekvencia. Na tento účel slúži 16-bit časovač Timer/ Counter1 s nastaveným výstupom generovaného hodinového signálu na PORTD PD4.

Obvod filtra predstavuje dostatočnú záťaž pre operačný zosilňovač a jeho výstupný odpor je dostatočne malý na pripojenie na vstupný port AD prevodníka. Výsledný odfiltrovaný výstupný signál z obvodu MAX7400 [6] má potom jednosmernú zložku 2,2V, čo predstavuje približný stred zo vstupného rozsahu AD prevodníka. Na indikáciu skreslenia slúži červená LED dióda pripojená na PORTA PA8. Vstupný signál je orezávaný AD prevodníkom zhora aj zdola. Indikátor porovnáva načítané hodnoty vo výsledkovom registri ADCH s hodnotami 0xFF a 0. Keď sa podmienka vyhodnotí ako pravdivá, je LED1 rozsvietená po dobu, kým nenastane ďalšie načítanie AD prevodníkom.

4.2. Realizácia DA časti modulu

Ďalšou súčasťou AD/DA modulu je možnosť spätnej rekonštrukcie digitálnych vzoriek signálu pomocou 2R- R paralelného prevodníka. Tento využíva celý 8-bit PORTC nastavený ako výstupný a je napájaný z logických úrovní. Posunutie hodnoty DA prevodníku teda môže prebehnúť v jednej inštrukcii mikroprocesora a na tento účel som využil už nastavený vzorkovací časovač Timer/counter0. Samotné posunutie hodnoty vzorky na PORTC sa vykonáva vo vektore prerušenia SIG_OUTPUT_COMPARE0.



Obr. 5. Schéma zapojenia DA časti modulu

DA prevodník bude interpretovať digitálne hodnoty v rozmedzí 0 až 5V rovných napájaniu vývojovej dosky. Najvyššiu váhu 8-bit hodnoty vzorky bude mať odpor R1 a najnižšiu zase R8. Hodnoty jednotlivých odporov je vhodné odmerať a spárovať pre zabezpečenie lineárnej prevodovej charakteristiky prevodníka. Keďže rekonštruovaný signál bude mať aj jednosmernú zložku pre vystredenie signálu medzi hranice 0 až 4V, je nutné túto zložku odstrániť pomocou kondenzátora C1. Toto je vhodné najmä kvôli vstupným parametrom obvodu rekonštrukčného filtra podľa špecifikácie.

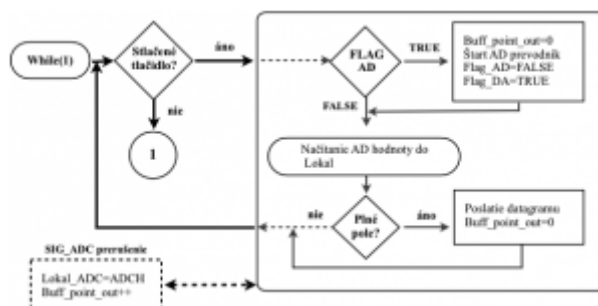
Pretože výstupný signál AD prevodníka obsahuje opakujúce sa frekvenčné komponenty v okolí vzorkovacej frekvencie a jej násobkom, je nutné zabezpečiť ich filtráciu. Na tento účel je tak, ako aj v AD časti, použitý obvod MAX 7401 [6] s Besselovou frekvenčnou charakteristikou. Medzná frekvencia 4kHz je nastavovaná tým istým výstupným hodinovým signálom z časovača Timer/Counter1, ako aj obvod pre antialiasing. Týmto je zároveň zabezpečená synchronizácia oboch filtrov. Na výstupe je možné regulovať výstupnú hlasitosť pomocou potenciometra R16.

5. Návrh programu

Program naväzuje spojenie s aplikáciou Matlab na konkrétnom porte so zadanou IP adresou počítača pomocou funkcie ARP_Resolutor(). Na prístup k vysielacím a prijímacím poliam a ako rozhranie aplikačnej a transportnej vrstvy TCP/IP protokolu boli na tento účel vytvorené funkcie UDP_receiver() a UPD_audio_stream(). Vzhľadom na to, že UDP datagramy budú posielené len v rámci lokálnej siete nie je významná požiadavka pre preposielanie chýbajúcich datagramov alebo ich zoradovanie na vstupnom pamäťovom poli v programe, keďže sa nejedná o väčší počet routrov, ktorými by boli preposielané. Tieto funkcie by sa ale stali nutnosťou v prípade komunikácie mimo lokálnu sieť. Vysielanie datagramov je aktivované stlačením jedným z dvoch tlačidiel na vývojovej doske.

5.1. Hlavný program

Najskôr sú do hlavného programu zahrnuté všetky potrebné knižnice TCP/IP protokolu. Taktiež sú deklarované všetky vytvorené funkcie a potrebné premenné obsahujúce adresy a čísla portov lokálneho a vzdialeného uzla, medzi ktorými bude prebiehať komunikácia, ako aj stavová premenná programu obsahujúca smerníky do polí a premenné stavu programu. Nakoniec sú inicializované jednotlivé polia veľkosti 500B na ukladanie vzoriek a program pokračuje iteráciou while(1).

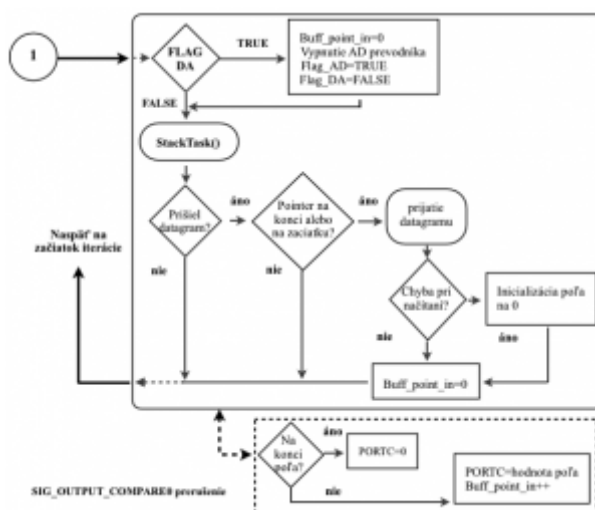


Obr. 6. Schéma bloku posielania datagramov

Na začiatku iterácie sa testuje, či bolo stlačené tlačidlo. Ak áno, program aktivuje vysielaciu časť. Ak ide o prvý prebeh blokom, je Flag_AD vynulovaný a v druhom prebehu už nebude časť kódu vo vetve TRUE (Obr. 6.) vyhodnotená. Taktiež je obnovená hodnota Flag_DA pre aktivovanie bloku príjmu v prípade pusteného tlačítka, vynulovaný ukazateľ do vysielacieho poľa Buffer_UDP_out a spustený AD prevod. Nová hodnota prevodu je v prípade vyhodnotenia prerušenia SIG_ADC načítaná do premennej Lokal_ADC a ukazateľ v poli je posunutý na ďalšiu pozíciu.

V bloku posielania sa potom nová premenná zapíše na nové miesto v poli. Ak prerušenie nenastane a v bloku posielania dôjde k požiadavke na načítanie novej

hodnoty, je hodnota v poli prepísaná tou istou hodnotou, takže nedojde k strate hodnoty. Nakoniec sa testuje, či je už pole plné. Ak už ukazateľ v poli ukazuje na poslednú hodnotu je obsah poslaný pomocou funkcie `UPD_audio_stream()`, nachádzajúcej sa v iterácii s určeným počtom opakovaní posielania premennou `SendTimeOutErr` pre prípad chyby načítania do pamäte modulu sieťového radiča.



Obr. 7. Schéma bloku prijímania datagramov

V prípade, že nie je ani jedno tlačidlo stlačené, je aktivovaný blok prijímania a prehrávania datagramov. (Obr. 7.) Tak ako v bloku posielania je aj v tejto časti testovaný prvý prebeh blokom. V tomto prípade je AD prevod vypnutý, aby prehrávanie nebolo rušené zbytočnými prerušeniami. Ak bol detekovaný prijatý UDP datagram pre túto aplikáciu, bude do premennej `Rx` uložená hodnota jeho dĺžky. Ak zároveň s detekciou datagramu nie je dokončené prehrávanie predošlého, volanie funkcie `UDP_receiver()` počká, kým sa obsah tohto datagramu prehrá do konca. Na tento účel slúži testovanie hodnoty ukazateľa v poli `Buff_point_in`.

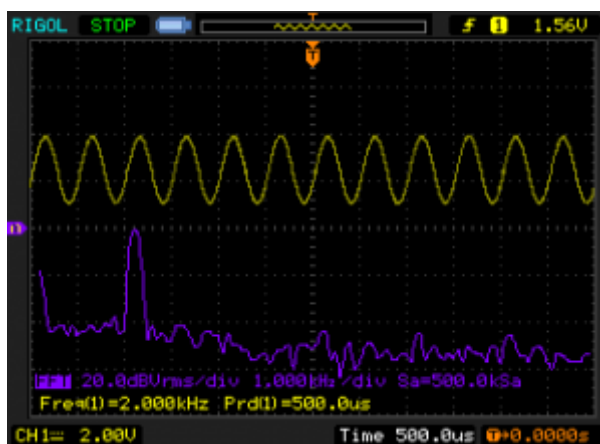
Až keď je dokončené prehrávanie a v pamäti sieťového radiča čaká prijatý datagram, sú uložené dáta v pamäti sieťového radiča funkciou `UDP_receiver()` nahraté do pola `Buffer_UDP_in`. V prípade, že funkcia `UDP_receiver()` vráti hodnotu `FALSE` nastala chyba načítavania alebo prišiel príliš veľký datagram, než aký je schopné vstupné pole načítať. Pole `Buffer_UDP_in` je potom nanovo inicializované na nuly funkciou `f_buffinit()`. Ukazateľ do pola je nakoniec po načítaní vynulovaný.

Samotné prehrávanie datagramov priradením jednotlivých hodnôt pola na `PORTC` je vykonávané v bloku vektora prerušenia `SIG_OUTPUT_COMPARE0`. V tomto bloku sa zároveň testuje, či je ukazateľ `Buff_point_in` v poli `Buffer_UDP_in` na konci. V prípade, že sa ukazateľ nachádza niekde v poli, je jeho hodnota po načítaní na port zväčšená. V opačnom prípade bude na port priradená hodnota `0x00`. Prehrávanie je dizajnované tak, aby keď dôjde k oneskoreniu nasledujúceho datagramu oproti predošlému, bola medzera medzi týmito datagrammi interpolovaná hodnotou `0x00`. V tomto prípade bude program akceptovať určitú chybu prehrávania. Taktiež bolo predpokladané dostatočne rýchle spracovávanie vstupných dát funkciou `UDP_receiver()`.

6. Výsledky

Pre testovanie funkčnosti modulu bol použitý upravený komunikačný program s

možnosťou načítania a spätného prehratia vzorky v rámci vývojovej dosky. Jednotlivé merania boli uskutočnené pomocou digitálneho osciloskopu Rigol DS1052E s pripojenou sondou 1x. Ako vstupný signál bol použitý harmonický tón frekvencie 2kHz a amplitúdy 50mV zo zvukovej karty počítača vygenerovaný pomocou funkcie `sound()` z Matlabu [5]. Signál sa potom pripojí na vstup AD modulu pomocou kábla.

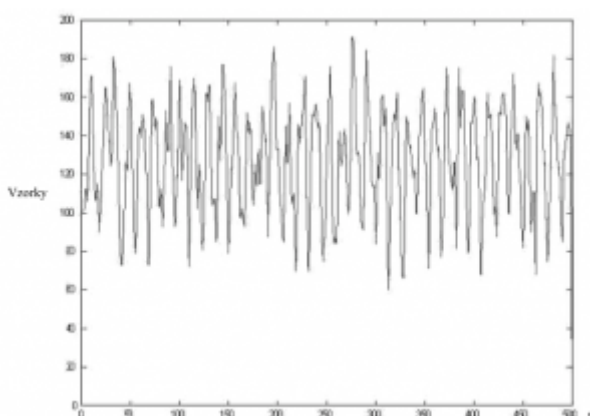


Obr. 8. Výstupný analógový signál DA časti modulu

Pri príliš veľkej amplitúde vstupného napätia signálu je rozsvietená LED na indikáciu skreslenia a je potom možné znížiť vstupné zosilnenie. Pomocou tejto LED sa dajú identifikovať aj veľmi krátke hodnoty napäťových špičiek, pri ktorých dochádza k prebudeniu vstupu. Frekvenčné komponenty nad 5kHz boli pomocou eliptických antialiasingových a interpolačných filtrov úplne odstránené. Frekvencie pod 300Hz boli potlačené menej, čo vyplýva z použitia RC článku filtra 1. rádu na vstupe.

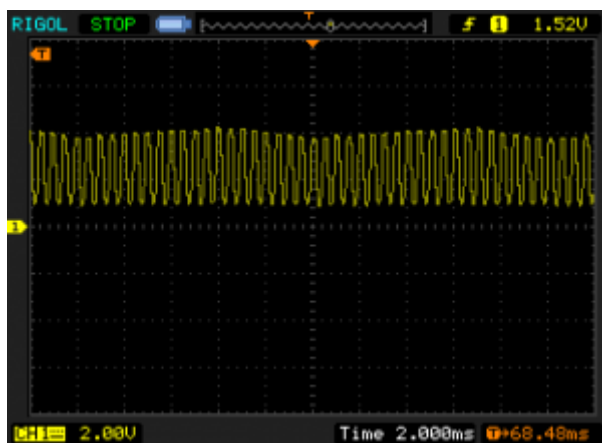
Aj keď neboli pri sluchovom teste počuteľné príliš veľké rozdiely, najlepšou kombináciou bolo použitie Eliptického filtra ako antialiasingového a Besselovho filtra ako interpolačného pre dosiahnutie dobrej počuteľnosti a nízkeho šumu. Pri teste bol použitý tienený 1W reproduktor s impedanciou 25Ω s pripojeným výstupným transformátorom.

Po stlačení tlačítka vývojovej dosky je aktivované načítavanie vstupného audio signálu a posielanie UDP datagramov. Pri posielaní sa vyšle celkovo 542 bytov a z toho je 42 bytov pre MAC, IP a UDP hlavičky TCP/IP stacku. Jednotlivé datagramy boli prijaté pomocnými funkciami do matice v Matlabe. Pomocou funkcie `plot()` boli potom vynesené jednotlivé hodnoty z matice s prijatými hodnotami. Obr. 9. predstavuje 31ms prehrávaného jedného audiodatagramu hudby.



Obr. 9. Zobrazenie obsahu prijatého audio-datagramu

Príjem UDP audio datagramov a prehrávanie DA časťou AD/DA modulu boli testované pomocou zostrojeného skriptu vysielajúceho UDP datagramy z Matlabu [5]. Obr. 10. zobrazuje prehrávaný obsah UDP datagramu s harmonickou funkciou pri frekvencii 2kHz a amplitúde s hodnotou 0x64 (100). Jednosmerná zložka bola nastavená na hodnotu 0x7D (125).

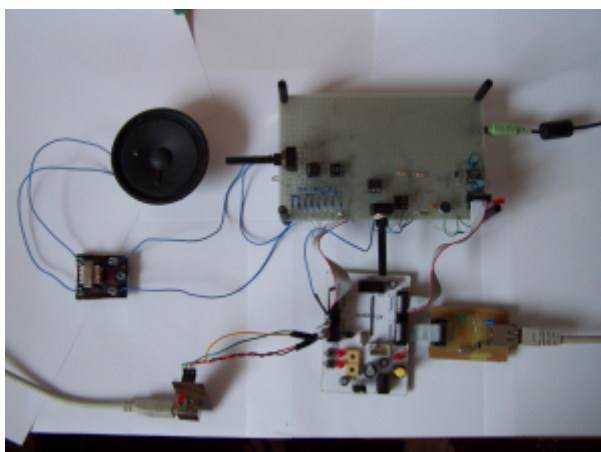


Obr. 10. Výstupný prehrávaný obsah datagramu

Pri prehrávaní medzi jednotlivými datagrammi dochádzalo k 4ms interpolácii nulou. Taktiež pri sluchovom teste bola táto pauza veľmi výrazná. Z tohto faktu vyplýva, že predpoklad doby spracovania datagramov funkciou `UDP_Receiver()` kratšej ako 62.5us nebol správny. Tento nedostatok sa dá odstrániť napríklad použitím kruhového bufera alebo striedavým prehrávaním a načítavaním dvoch polí, čo by prekrylo čas potrebný na načítanie datagramu.

7. Záver

Táto práca demonštruje možnosti realizácie ethernetovej VOIP duplexnej vysielačky pomocou 8-bit mikropočítača Atmega32 a periférneho sieťového modulu s obvodom ENC28J60. Funkčnosť návrhu bola overená pomocou meraní napätia s osciloskopom, FFT analýzou a tiež sluchovým testom. Taktiež bol prezentovaný návrh, realizácia a navrhnuté zlepšenia programu pre riadenie modulu a základnú duplexnú ethernetovú komunikáciu s aplikáciou Matlab. Podobné zariadenie by mohlo nájsť uplatnenie napríklad ako digitálny telefón vo vlakových súpravách, pripojený na spoločnú ethernetovú zbernicu.



Obr. 11. Ukážka zariadenia

8. Literatúra

1. Cascad, J., "Sams Teach Yourself TCPIP in 24 Hours", 450s, Sam Pub., ISBN:0-67-32565-9, 2003
2. Adams, R., et. al., "DAC ICs: How Many Bits is Enough?", AN327, Analog devices, 1991
3. AN928 "Filter Basics: antialiasing", Maxim, 2002
<http://www.maxim-ic.com/app-notes/index.mvp/id/928>
4. "8-bit AVR Microcontroller ATmega32", datasheet, Rev. 2503O-07/09, Atmel Corporation 2010
5. The MathWorks, Inc. Instrument control toolbox, 2010
<http://www.mathworks.com/access/helpdesk/help/toolbox/instrument/>
6. "8th order lowpass Bessel/elliptic switched capacitor filters ", datasheet, Maxim ,Rev. 1; 6/99, 19-4788
7. Križan, M, práca Švoč 2009 "Riadenie a monitorovanie prostredníctvom počítačovej siete LAN"

Spoluautorom práce je Ing. Zdenko Brezovič, Katedra rádioelektroniky, Fakulta elektrotechniky a informatiky STU Bratislava
