UNIVERZITA PARDUBICE

DOPRAVNÍ FAKULTA JANA PERNERA

KATEDRA ELEKTROTECHNIKY, ELEKTRONIKY A ZABEZPEČOVACÍ TECHNIKY V DOPRAVĚ

MĚŘENÍ NF SIGNÁLŮ POMOCÍ A/D PŘEVODNÍKU S VYSOKÝM ROZLIŠENÍM

BAKALÁŘSKÁ PRÁCE

AUTOR PRÁCE: **Martin Šafařík**

VEDOUCÍ PRÁCE: **Ing. Jan Mrkvica, Ph.D.**

2007

UNIVERSITY OF PARDUBICE

JAN PERNER TRANSPORT FACULTY

DEPARTMENT OF ELECTRICAL AND ELECTRONICAL ENGINEERING AND SIGNALLING IN TRANSPORT

LOW FREQUENCY SIGNALS MEASURMENT USING HIGH RESOLUTION A/D CONVERTER

BACHELOR WORK

AUTHOR: **Martin Šafařík**

SUPERVISOR: **Ing. Jan Mrkvica, Ph.D.**

2007

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 20.5.2007

Martin Šafařík

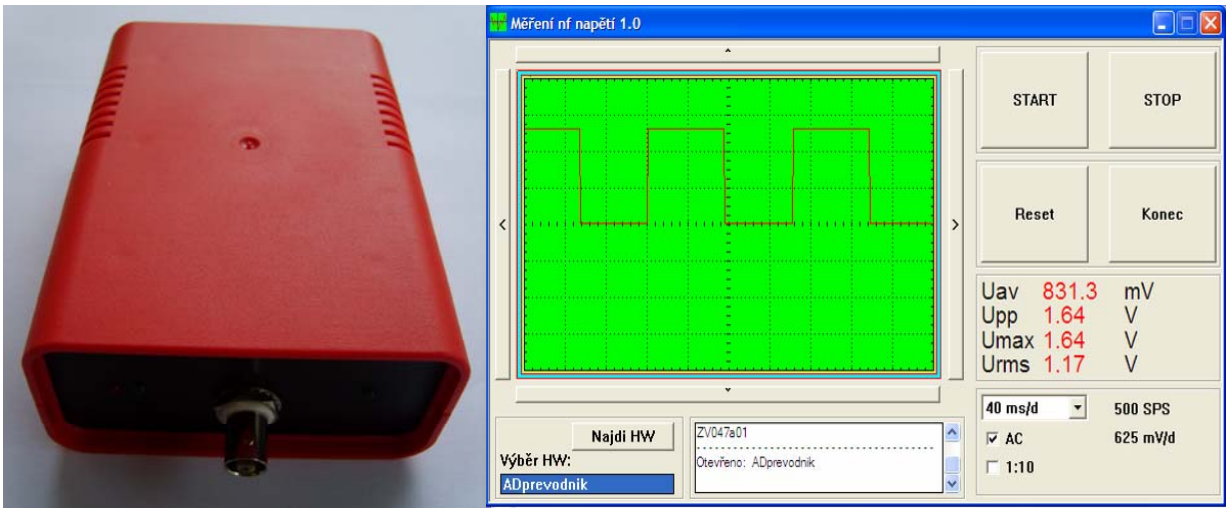
Poděkování

Zde chci poděkovat společnosti Retia a.s. za veškerou možnou podporu a především vedoucímu této práce Ing. Janu Mrkvicovi, Ph.D. za důležité připomínky, rady, nápady a další pomoc při práci na vytvořeném zařízení, jenž je předmětem této práce.

Souhrn

Zařízení pro zjišťování průběhů nízkofrekvenčních signálů v čase jsou v určitých situacích velice užitečná. Právě takové zařízení je popisováno na následujících stránkách. Funkce zařízení je založena na součinnosti A/D převodníku, mikrokontroléru a počítače.

Nejdříve jsou v teoretickém rozboru uvedeny základní informace o A/D převodnících, jejich vlastnostech a některých jejich parametrech, protože A/D převodník určuje z velké části vlastnosti vyrobeného zařízení. Následně jsou konkrétně popsány hlavní části, které byly použité v zapojení. Jsou to: A/D převodník AD7675, USB/UART konvertor FT232BM a mikrokontrolér ATmega16. Dále je uveden popis hardware a zdůvodnění použití jednotlivých součástí. Je také uveden princip práce zařízení a způsob jeho ovládání pomocí řídícího programu. Nakonec je zařazen i popis software, který je rozdělen na dvě části. Jako první je popsán program pro mikrokontrolér, pak následuje popis ovládacího programu pro počítač. V příloze lze nalézt motiv plošného spoje, fotografie přístroje a další příklady měření.

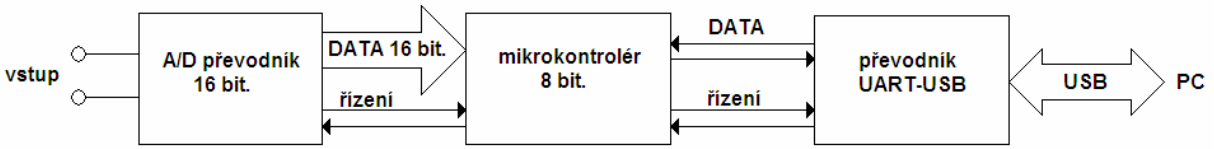


Obsah

Úvod

Pro zjištění časového průběhu napětí nebo např. proudu používáme osciloskopy. Tyto přístroje mají velice široké možnosti uplatnění. Jejich velikou nevýhodou je však vysoká pořizovací cena, zejména u digitálních osciloskopů. Cílem této práce je proto vytvoření zařízení, jenž by bylo schopno zobrazit průběh měřeného napětí. Mělo by tedy umožňovat měření amplitudy přivedené elektrické veličiny a u střídavých průběhů i měření frekvence, alespoň pomocí rastru na obrazovce. Maximální zobrazitelná frekvence napětí nebude příliš vysoká, protože by se výrazně zvýšila složitost i celkové náklady na zařízení.

Celé zařízení se skládá z těchto hlavních součástí – A/D převodník, mikrokontrolér a konvertor UART-USB. Zobrazování naměřených hodnot je realizováno na monitoru počítače. Mikrokontrolér řídí ostatní prvky zařízení. Pomocí přesného A/D převodníku je měřený signál ovzorkován a jeho amplituda kvantována. Naměřená data jsou načítána do mikrokontroléru a prostřednictvím konvertoru posílána do počítače. Spojení celého zařízení s počítačem je realizováno přes sběrnici USB. Ovládání přístroje je prováděno speciálním programem. Obslužný software umožňuje nejen spouštět a zastavovat měření, ale také měnit vzorkovací frekvenci, v několika krocích. Řídící příkazy programu jsou zpracovávány a vykonávány mikrokontrolérem. Způsob realizace hardwaru zařízení je vidět na obr. 1.



Obr. 1 Blokové schéma popisovaného zařízení

Po příchodu povelu z počítače je A/D převodníkem zjištěna amplituda asi čtyř set vzorků napětí ze vstupního signálu. Vzorky jsou patřičně kódovány a výsledná data sériově vyslána do počítače, kde jsou v určitém měřítku naměřené hodnoty zobrazeny. Vzorkovací rychlost je nastavována pomocí povelů z počítače. Rychlost přenosu naměřených dat do počítače je mnohem nižší než rychlost vzorkování, proto je najednou odebrán blok čtyř set vzorků, který se uloží do rychlé paměti, později je tento blok odeslán do počítače. Zobrazení naměřeného průběhu je realizováno pomocí postupného spojování naměřených hodnot přímkami, bez aproximace. Tvar výsledné křivky by pak měl odpovídat naměřenému signálu v určitém měřítku. Přesnost zobrazování amplitudy průběhu je jeden pixel. Měření frekvence vstupního signálu pomocí rastru však takové přesnosti nedosahuje.

1. Teoretický rozbor řešení

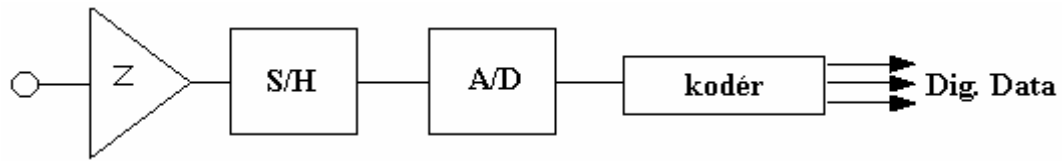
Tato kapitola obsahuje základní informace o analogově-digitálních převodnících a pro některé typy je uveden i popis jejich funkce. Přesnost celého zařízení je totiž dána především použitým převodníkem. Nevhodným výběrem tohoto převodníku by bylo celé zařízení znehodnoceno. Proto je tomuto obvodu věnována samostatná kapitola. A/D převodník je zařízení, které je velmi náchylné k rušení. Při výběru je nutné vyhledat takový typ, který bude vyhovovat našim požadavkům. Aby byl zřejmý důvod výběru použitého A/D převodníku, je nutné o těchto zařízeních znát základní informace. To je obsaženo v následující kapitole.

1.1 Obecný popis A/D převodníků

Analogově-digitální převodníky jsou zařízení, která slouží pro digitalizaci analogového vstupního signálu. V textu je označujeme zkratkami A/D, A/Č nebo ADK či ADC (Analog to Digital Converter). Často se u těchto převodníků, i obecně v číslicové technice, používají dvě zkratky: MSB = bit s nejvyšší váhou (Most Significant Bit), LSB = bit s váhou nejnižší (Least Significant Bit). Jedním ze stavebních prvků některých A/D převodníků jsou operační zesilovače. Z těchto zesilovačů mohou pak být složeny základní sub-obvody těchto převodníků. Jsou to speciálně *invertory,* tj. zesilovače s inverzním výstupním napětím, (se zesílením -1), dále *komparátory*, jež svůj výstup mění podle velikosti napětí na obou vstupech a posledním prvkem, sloužícím k impedančnímu oddělení sousedních prvků v obvodu je *napěťový sledovač*. V neposlední řadě můžeme pomocí operačního zesilovače amplitudově upravovat vstupní signál. Bližší informace v [7] nebo [9].

Některé A/D převodníky používají pro nastavení váhových proudů do komparátorů váhové rezistory. Přesnost převodníků pak závisí především na přesnosti těchto rezistorů. Ve snaze vyhnout se této situaci se používají principy, u kterých už na přesnosti rezistorů příliš nezáleží.

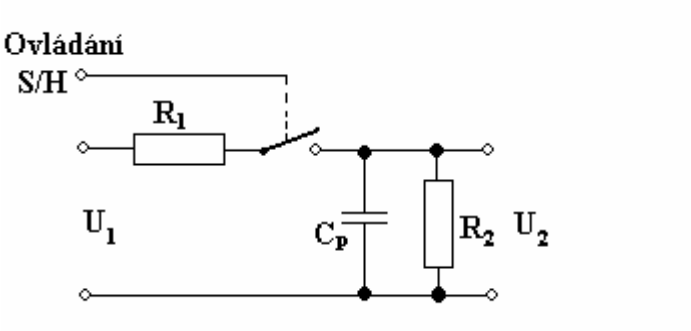
Obecný A/D převodník (obr. 2) si můžeme představit jako obvod obsahující: vstupní *zesilovač* (či *dělič)*, vzorkovací obvod *S/H* (Sample and Hold), vlastní *A/D převodník* a *kodér*.



Obr. 2 Obecné schéma A/D převodníku

Vstupní zesilovač amplitudově upraví vstupní analogový signál tak, aby vyhovoval další prvkům v obvodu. Vzorkovací část má dvě funkce. Jednak načte okamžitou hodnotu vstupního napětí (stav Sample), ale dokáže také tuto hodnotu udržet (stav Hold) tak dlouho, dokud nebude provedena její konverze na digitální data. Ve vlastním A/D převodníku se mění navzorkovaný vstupní signál na digitální výstup. Princip převodu je u každého typu převodníku jiný. V posledním výstupním obvodu A/D převodníku se digitalizovaným hodnotám přiřadí některý z číselných kódů, nejčastěji kód binární. Data se dále upravují podle požadavků dalšího zpracovaní, viz [9].

Princip obvodu S/H je znázorněn na obr. 3. Stav sledování (Sample) je realizován sepnutím spínače a tedy přivedením vstupního napětí U_1 na paměťový kondenzátor C_p. Stav pamatování (Hold) je potom dán rozepnutím spínače. Výstupní napětí U_2 je pak rovno napětí na paměťovém kondenzátoru. Jako spínače lze používat např. tranzistory. Sériovým zapojením několika vzorkovacích obvodů s různými kapacitami paměťových kondenzátorů se velice zkvalitní vlastnosti vzorkování.



Obr. 3 Princip S/H obvodu

Při vzorkování měřeného signálu je nutno dodržet Shannon-Kotělnikovu větu, aby nedocházelo k aliasingu, tedy k překrývání kmitočtových spekter signálu. Vzorkovací věta ve svém důsledku říká, že vzorkovací frekvence musí být alespoň dvakrát vyšší než nejvyšší frekvence obsažená ve spektru měřeného signálu.

1.2 Parametry A/D převodníků

Při základním výběru A/D převodníku se budeme rozhodovat podle základních kritérií. Nejdříve bychom měli proto uvažovat o těchto parametrech a pokud je vybraný typ splňuje, můžeme začít sledovat i detailnější specifikace.

ZÁKLADNÍMI PARAMETRY PŘI VÝBĚRU A/D PŘEVODNÍKU JSOU TYTO:

- **Poměr potenciál/cena**
- **Vzorkovací rychlost a šířka pásma**
- **Počet bitů = rozlišení**
- **Dostupnost na trhu**
- **Organizace výstupních dat –** sériová, paralelní, ve dvojkovém doplňku, standardně
- **Provedení vstupu** – jeden vstup a zem (Single Ended) nebo rozdílově (Diferencial)
- **Počet vstupních kanálů**
- **Rozsah vstupního napětí a unipolární nebo bipolární možnost měření**
- **Pouzdro převodníku, nutnost připojení krystalu, napěťové reference**
- **Příkon, napájení** – nejen velikost napětí, ale také jeho unipolárnost nebo bipolárnost
- **Technologie provedení –** TTL, CMOS, …

Reálný převodník je zdrojem mnoha rušivých signálů – šumů, obsahuje různé nelinearity a nestability. Proto nás zajímají i další jeho parametry, a to jak statické (odstup signál/šum S/N, efektivní počet bitů ENOB, dynamický rozsah DR_{AD}, integrální a diferenciální nelinearita INL, DNL), tak i dynamické (šířka pásma BW, doba odběru vzorku T_a, fázová nestabilita - jitter, harmonické zkreslení THD, atd.).

Odstup signál/šum S/N (Signal/Noise, též SQRN), vytvářený vlastním převodem, lze pro harmonický signál s dostatečnou amplitudou vyjádřit v dB takto:

$$S/N = 1{,}76 + 6{,}02 \cdot n, \tag{1}$$

kde n je počet bitů ideálního převodníku.

Bude-li však vzorkovaný signál kmitočtově omezený v pásmu BW, menším než polovina vzorkovacího kmitočtu - fv/2, pak do pravé strany rovnice (1) musíme přidat ještě výraz $10*\log(f_v/2BW)$. Tento člen vyjadřuje dodatečný zisk zpracování, vznikající při vzorkování kmitočtem vyšším, než který vyplývá ze vzorkovací věty.

ENOB (Efficient Number Of Bits) je dalším dobře charakterizujícím parametrem A/D převodníku. Je to vlastně efektivní počet bitů převodníku (efektivní rozlišitelnost převodníku) a v podstatě nám říká, kolik bitů můžeme skutečně plně využít. Toto číslo se určuje pro plné vybuzení harmonickým signálem.

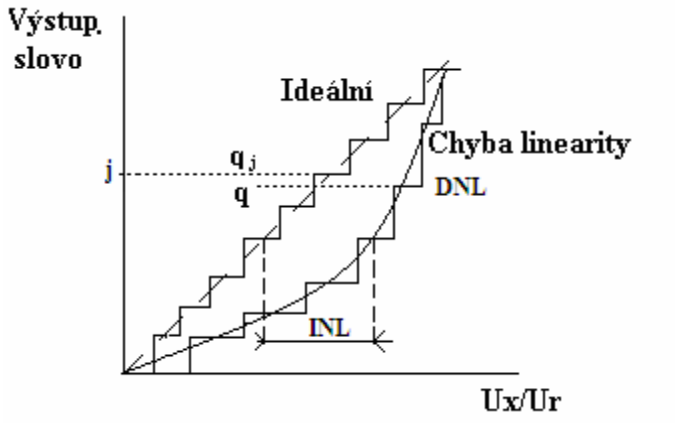
Počet efektivních bitů bude vždy menší než skutečný počet bitů, protože všudypřítomný kvantizační šum tuto hodnotu degraduje. Parametr ENOB vypočítáme ze vztahu (1), kde je však označen jako *n,* přičemž za S/N nyní dosazujeme odstup signál/šum celého hardwaru převodníku, tedy např. i integrovaného vzorkovače.

Pokud ENOB vynásobíme konstantou 6,02, pak obdržíme dynamický rozsah převodníku DR. Odstup signál/šum + zkreslení označujeme jako SINAD.

Nelineárním zesílením zesilovače, vznikají dvě význačné chyby. Integrální nelinearita INL (Intergral Non Linearity) je určena odchylkami středů kvantovacích úrovní ideálního a reálného převodníku, viz obr. 4. Diferenciální nelinearita DNL (Diferenciál Non Linearity) je určena rozdílem kvantovací úrovně q_j reálného a úrovně q ideálního převodníku pro j-té slovo. Důsledkem chyby DNL je v podstatě to, že nejmenší kvantovací krok není roven LSB. Podle obr. 4 platí:

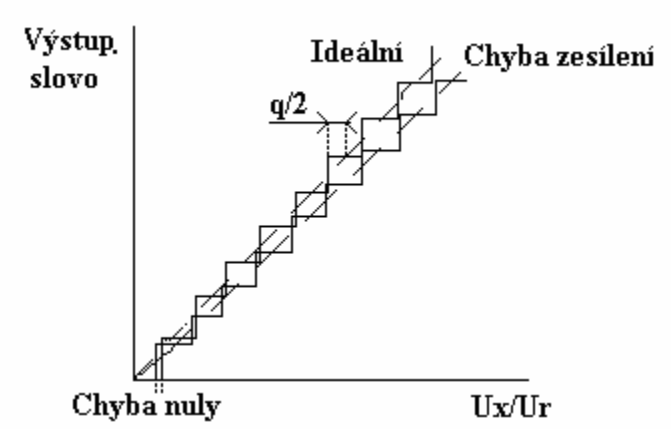
$$DNL = \frac{q_j - q}{q} \tag{2}$$

kde q_j je ideální amplituda j-tého slova, q je skutečná-chybná amplituda j-tého slova.



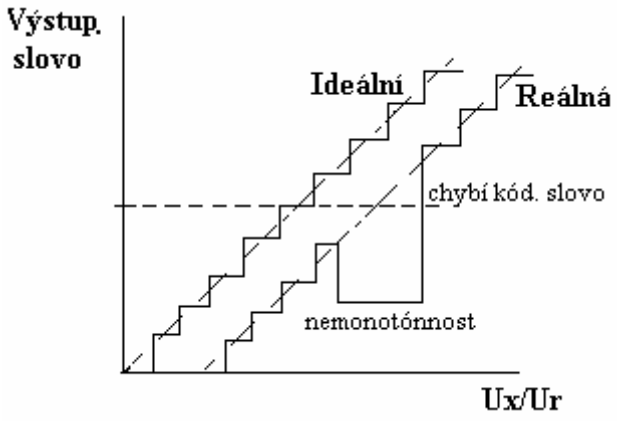
Obr. 4 Převodní charakteristiky:
Ideální a zatížená chybami DNL a INL

Významné statické chyby jsou podle obr. 5: chyba nuly, chyba zesílení a chyba kvantování, nabývající maximální hodnoty q/2.



Obr. 5 Chyba zesílení, chyba nuly a chyba kvantování

Další možná nepřesnost v převodu signálu je nemonotónnost a z ní plynoucí ztráta kódového slova. Vyobrazení těchto chyb je na obr. 6.



Obr. 6 Chybějící kódové slovo a nemonotónnost

Harmonické zkreslení převodníku až do n-té harmonické složky počítáme takto:

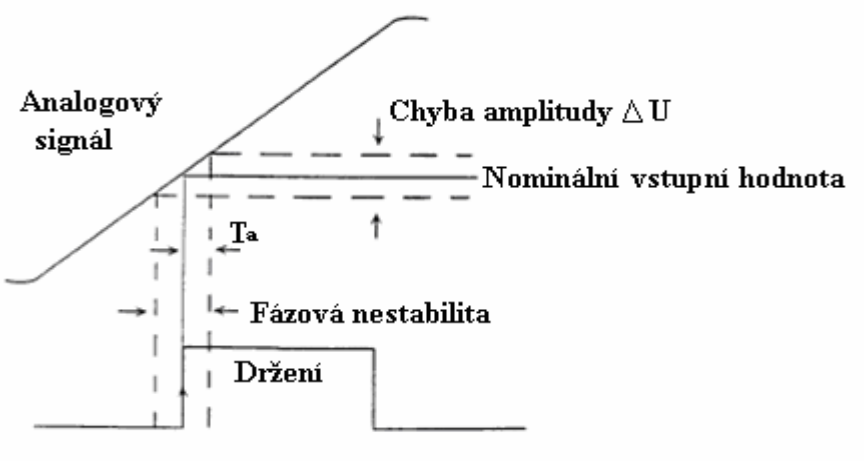
$$THD = \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + \cdots + V_n^2}}{V_1}, \tag{3}$$

kde V jsou amplitudy vyšších harmonických.

Analogová šířka pásma převodníku končí tam, kde amplitudově poklesne výstupní signál o 3dB, tedy přibližně na úroveň 70% amplitudy ve středu charakteristiky. Doba vzorkování $T_{a,}$ podle obr 7. způsobuje při odběru jednoho vzorku chybu:

$$du_a = \frac{du}{dt} \cdot T_a \tag{4}$$

kde první člen na levé straně je změna sledovaného signálu a T_a je doba odběru vzorku.



Obr. 7 Chyba při vzorkování

Maximální měřitelný vstupní kmitočet je při dodržení všech podmínek dán výrazem:

$$f_m \leq \frac{1}{N.2.\pi.T_a} \tag{5}$$

kde N je počet bitů daného A/D převodníku a T_a je doba odběru vzorku.

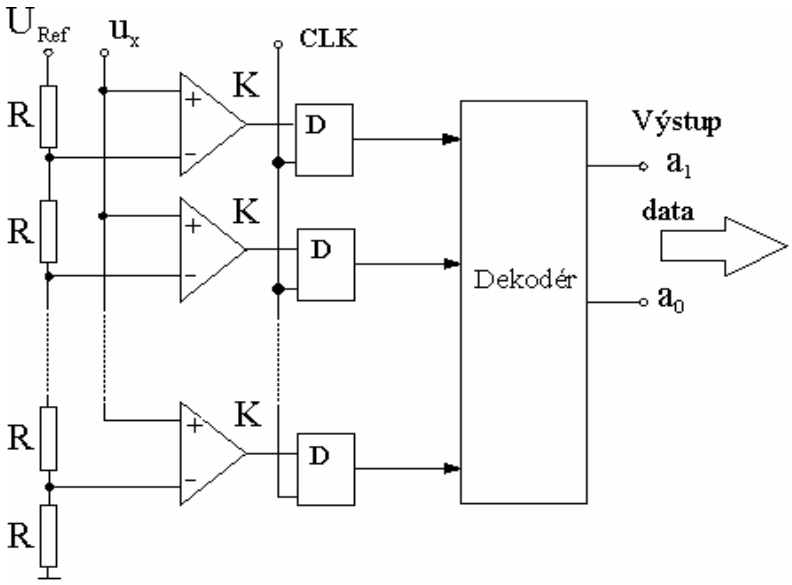
1.3 Některé typy A/D převodníků

1.3.1 Paralelní (komparační, flash) převodníky

Paralelní převodníky jsou velmi rychlé. Proto převodník postavený na tomto základě bude asi nejvhodnější pro zařízení pracující s vyššími kmitočty. Flash převodníky jsou zatím vyráběny nejvíce jako dvanáctibitové, pro vyšší frekvence však pouze jako šestibitové.

Vstupní signál je kvantován v paralelně zapojených komparátorech s ekvidistantně odstupňovaným referenčním napětím. Postupně je tedy na rezistorech R napětí $U_r/2$, $U_r/4$, $U_r/8$, atd. Převodník s n-bity musí být sestaven z 2^n-1 komparátorů. Právě nutnost použití velkého počtu komparátoru omezuje rozlišení převodníku. Za komparátory jsou někdy umístěny klopné obvody typu D. Doba zpoždění zápisu stavů komparátorů do paměti představuje dobu odebrání vzorku T_a. Výstup komparátorů je v Grayově kódu. Používáme dekodéry, umožňující zvýšit vzorkovací kmitočet pomocí principu pipe line, tj. současné konverzi jednoho vzorku a dekódování předešlého. Další zvýšení frekvence vzorkování spočívá v paralelním zapojení dvou převodníků, kde první vzorkuje signál a druhý současně dekóduje předešlý vzorek, v dalším taktu si funkce vystřídají.

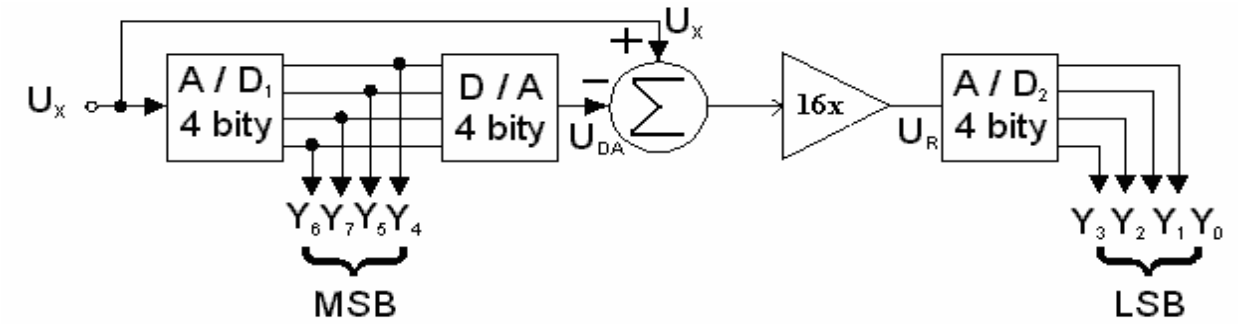
Typické zapojení tohoto převodníku je znázorněno na obr. 8.



Obr. 8 Paralelní převodník

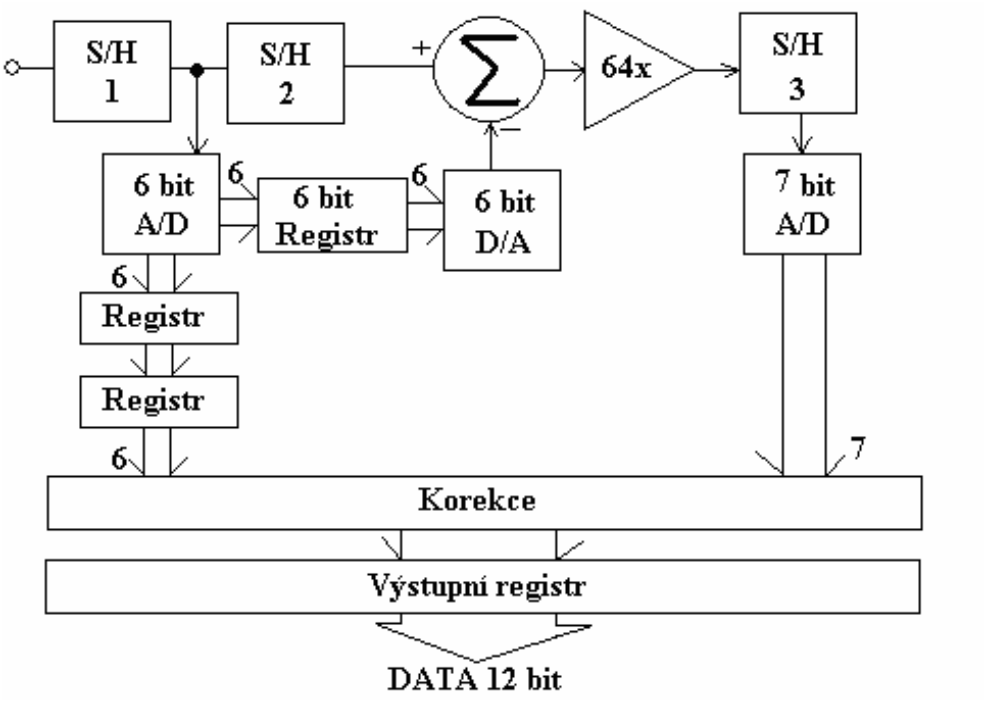
1.3.2 Odečítací (paralelně kaskádní) převodníky

Osmibitový převodník na obr. 9 je složen ze sčítačky, zesilovače, dvou A/D převodníků čtyřbitové architektury a jednoho čtyřbitového D/A převodníku. Principielně pracuje tak, že nejdříve určí hodnoty nejvýznamnějších čtyř bitů, ty pomocí D/A převodníku zpětně převede na analogový signál. Tento rekonstruovaný signál se následně odečte od původního vzorkovaného signálu. Zbytkový signál z odečítacího obvodu je 16x zesílen (256x pro 16b ADC) a přiveden na druhý A/D převodník, jenž má stejné parametry jako vstupní převodník. Výstupy obou převodníků se sloučí a vznikne osmibitový datový výstup. Nemonotónnost převodu se potlačuje překrýváním rozsahů obou A/D převodníků. Jsou vyráběny s rozlišením až 16 bitů. Nejčastěji jsou oba A/D převodníky komparačního typu.



Obr. 9 Osmibitový odečítací převodník

Moderní odečítací převodníky (obr. 10) nepoužívají už flash technologii, ale využívají stupně MagAmp, což je speciální typ diferenciálního zesilovače. Funkce těchto převodníků spočívá také v použití více obvodů S/H ve snaze zvýšit vzorkovací kmitočet. Dvanáctibitový převodník používá jeden šestibitový D/A a dva A/D převodníky, z nichž první je šestibitový a druhý sedmibitový, jeden bit navíc je využit pro korekce chyb. Princip je zpočátku stejný jako u klasického odečítacího převodníku, ale mezi šestibitovým A/D a D/A převodníkem je zařazen registr. Po prvním šestibitovém převodu a uložení výsledku je opět signál zpětně převeden na analogový. Pak je tento signál odečten od původního. Druhý S/H, zpožďující výstup prvního, je zařazen před sčítačkou. Zbytek analogového signálu je zesílen a přiveden na sedmibitový D/A převodník, ale přes další obvod S/H, jenž odstraňuje případné rušivé napěťové špičky. Za prvním šestibitovým A/D převodníkem jsou před výstupním dekodérem a registrem chyb zařazeny dva registry, to proto, aby byla horní i dolní šestice bitů výsledku časově správně nakombinována. Znázorněný typ A/D převodníku je dvoustupňový, ale používané jsou i převodníky čtyřstupňové, jednotlivé stupně jsou pak zapojeny za sebou.



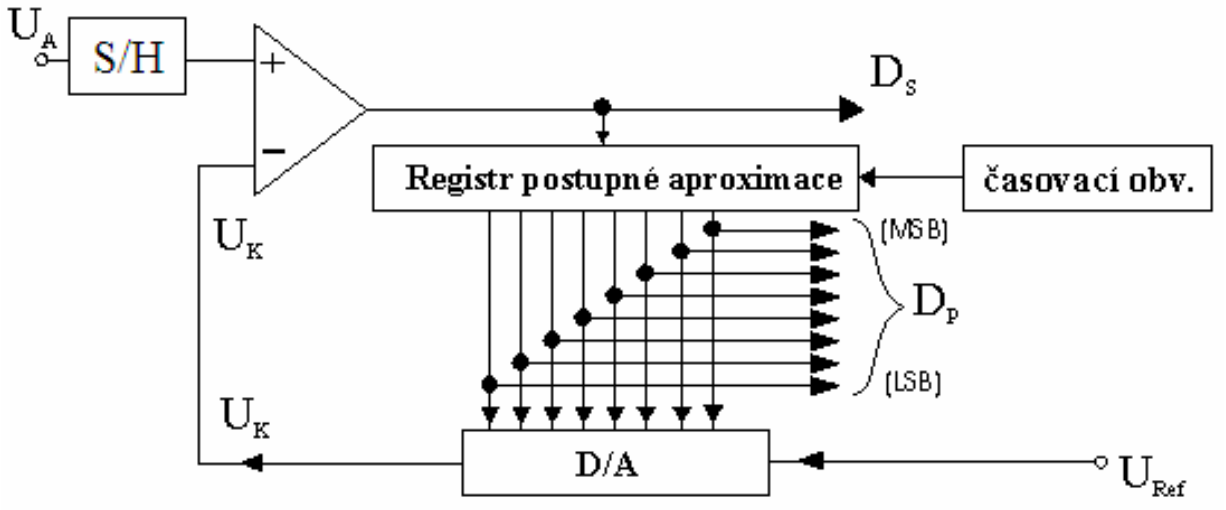
Obr. 10 Odečítací 12b A/D převodník

1.3.3 Aproximační převodníky

Obvodově se tyto typy skládají z D/A převodníku, registru postupné aproximace, komparátoru, obvodu S/H a časovacího obvodu. Měřené napětí je přes obvod S/H vedeno na jeden vstup komparátoru. Princip převodu pak spočívá v postupném připojování váhových velikostí napětí U_k, které jsou přiváděny na druhý vstup komparátoru. Registr postupné aproximace SAR (Successive Aproximation Register) je většinou tvořen klopnými obvody typu D nebo JK.

Nejprve se v registru postupné aproximace nastaví nejvyšší bit (MSB) na hodnotu H (High, logická jednička). D/A převodník tuto digitální informaci převede na odpovídající analogové napětí. Poměr mezi digitálním a analogovým napětím vystupujícím z D/A převodníku je dán napěťovou referencí. Takto vzniklý signál je přiveden na komparátor, kde je porovnána jeho amplituda s měřeným napětím. Bude-li amplituda měřeného napětí nižší, pak MSB v registru postupné aproximace bude nulován. V opačném případě mu bude ponechána hodnota H. Ve druhém taktu je nastaven druhý nejvyšší bit na hodnotu H a porovnání s měřeným napětím se bude opakovat. První takt při měření je vyhrazen pro nulování registru, v posledním taktu je z SAR vyslána potvrzující informace o dokončení převodu, tzv. signál EOC (End Of the Conversion).

Na obr. 11 zobrazený převodník má výstupní data označena D_p (paralelní výstup). Má však i výstup D_s (sériový výstup) pro kaskádní napojení na další převodníky stejného typu.



Obr. 11 Aproximační převodník

Přesnost aproximačních převodníků závisí především na vnitřním D/A převodníku, který jsou tvořen maticí tenkovrstvých rezistorů. Odpor rezistorů však není ideálně stálý, je závislý především na teplotě, ale i na dalších veličinách. Proto začaly být použité rezistory postupně kompenzovány pomocí laserového dostavování. Toto však vedlo k neúměrnému zvýšení nákladů na výrobu těchto převodníků. Bylo nutné změnit a hlavně zlevnit princip použitých D/A převodníků.

1.3.4 Aproximační převodníky s přerozdělováním náboje

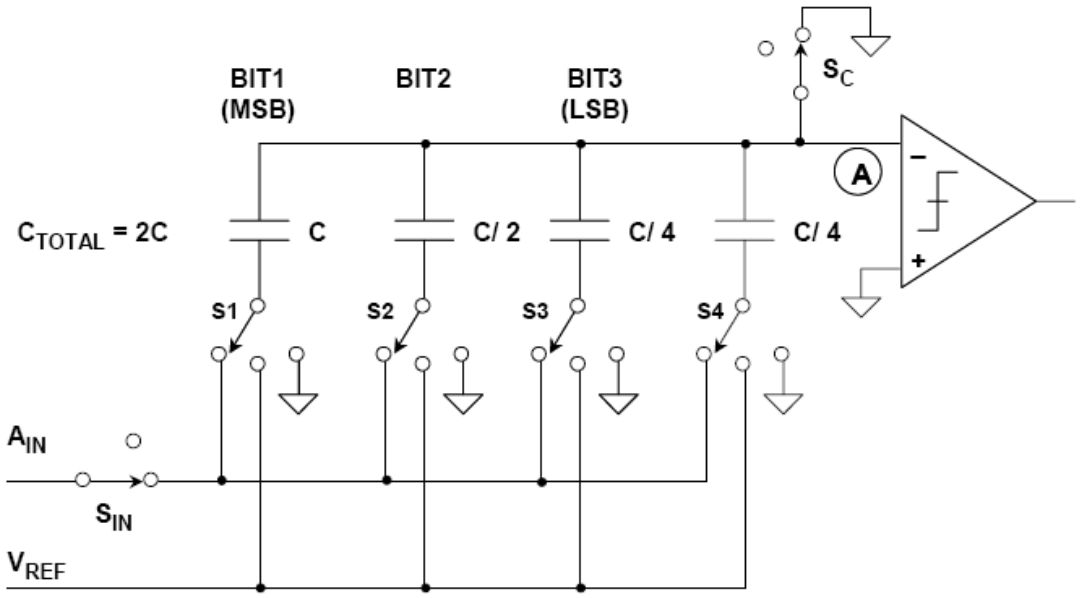
Moderní aproximační A/D převodníky pracují téměř výhradně na principu přerozdělování náboje (charge redistribution). Vnitřní D/A převodník je tvořen maticí váhových kondenzátorů (viz obr. 12), přičemž kapacita každého dalšího je dvojnásobná oproti kapacitě předchozího. Každý kondenzátor je ovládán samostatně pomocí tranzistorového spínače.

Funkce převodu napětí je opět velice jednoduchá. Nejprve jsou kondenzátory zcela vybity. Potom jsou spínače všech kondenzátorů přepnuty do vyznačené polohy. Zakreslená situace je stav sledovací. Součet nábojů všech kondenzátorů je pak v každém okamžiku úměrný měřenému napětí Uin. Přes jednoduchý vztah mezi nábojem a kapacitou:

$$Q=C*U \tag{6}$$

kde pro každý kondenzátor je: U – napětí na jeho deskách, Q - náboj, C – kapacita

Další fází je stav držení, při které jsou rozepnuty spínače Sc a Sin. Kondenzátory jsou tedy odpojeny od měřeného napětí Ain. Přepínače u kondenzátorů jsou pak přehozeny do polohy, ve které jsou kondenzátory spojeny se zemí. Napětí v bodě A bude tedy -Ain, tedy inverzní k napětí měřenému. Kondenzátor představující MSB je pak připojen na referenční napětí, které má amplitudu o hodnotu nejnižšího bitu vyšší, než je maximální měřitelné napětí Ain_{max}. Napětí na MSB kondenzátoru je nyní Vref / 2 – Ain. Komparátor pak rozhodne, zda tento spínač zůstane sepnut, tedy jestli je napětí na kondenzátoru MSB nižší nebo rovno polovině referenčního. Po této operaci se pokračuje stejně i na dalších dvou bitech (na obrázku je tříbitový typ). Na konci převodu jsou kondenzátory opět vybity vhodnou kombinací poloh spínačů a přepínačů. Důvodem připojení druhého LSB kondenzátoru je rovnost celkové kapacity matice kondenzátorů hodnotě 2C (C+C/2+C/4+C/4). Důsledkem je, že na kondenzátorech se objeví správně odstupňované napětí i při manipulaci s přepínači u jednotlivých kondenzátorů.



Obr. 12 Aproximační převodník s přerozdělováním náboje

1.3.5 Integrační převodníky

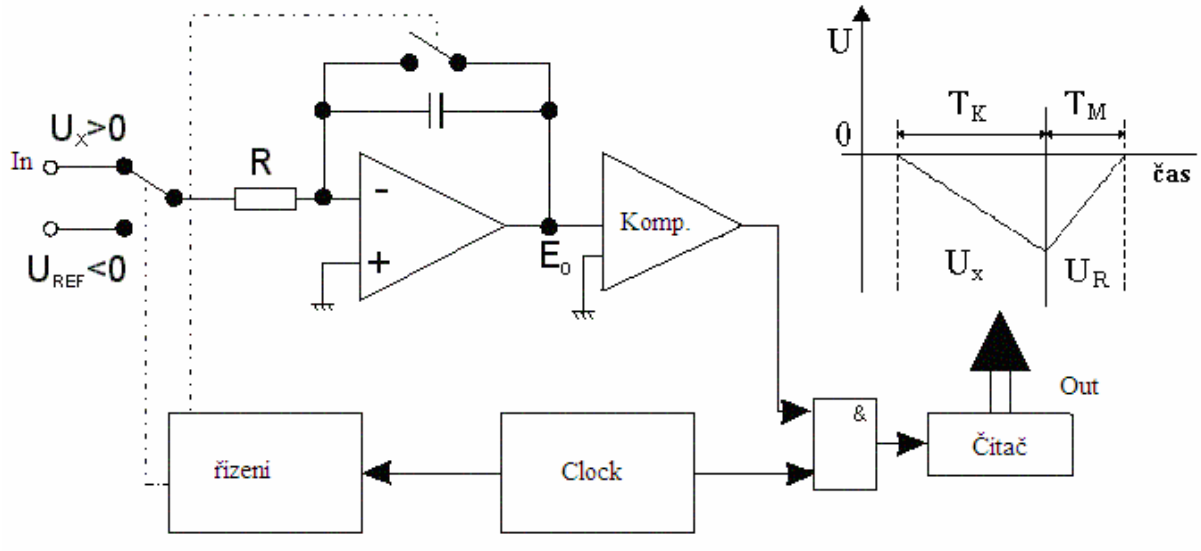
Mechanismus převodu spočívá v převedení měřeného napětí na časový interval. Po tento čas je otevřeno hradlo, přes které prochází úzké pulsy stejné frekvence a jejich počet je úměrný měřenému napětí. Toho se hojně využívá u digitálních měřicích přístrojů. Tento převod je přesný, ale není příliš rychlý.

S jednotaktní integrací

Používány jsou zde dva komparátory, integrátor, hradlo a čítač. Překročí-li výstup integrátoru Ui pevné napětí Ur, první komparátor otevře hradlo do čítače, jenž čítá do doby, kdy se rovnají napětí měřené a napětí z prvního komparátoru. V tento okamžik první komparátor hradlo uzavře. Počet impulsů prošlých přes hradlo do čítače je přímo úměrný vstupnímu napětí.

S dvoutaktní integrací

V prvním taktu je pro pevně daný časový interval T_k naintegrováno měřené napětí Ux. Ve druhém taktu je vstup přepojen na stálé, dostatečně velké referenční napětí opačné polarity a toto napětí U_r je odintegrováno. Čas odintergrace T_M je přímo úměrný původně naintegrovanému měřenému napětí. Čím je Ux menší, tím má křivka jeho integrace menší směrnici a za pevný čas T_k tudíž dosáhne menší úrovně naintegrování, a tedy rychlejší odintergrace pevným opačným referenčním napětím U_r. Tímto principem lze dosáhnout až 16 bitové přesnosti. Základní schéma je na obr. 13.



Obr. 13 Integrační převodník s dvoutaktní integrací

S vícestupňovou integrací

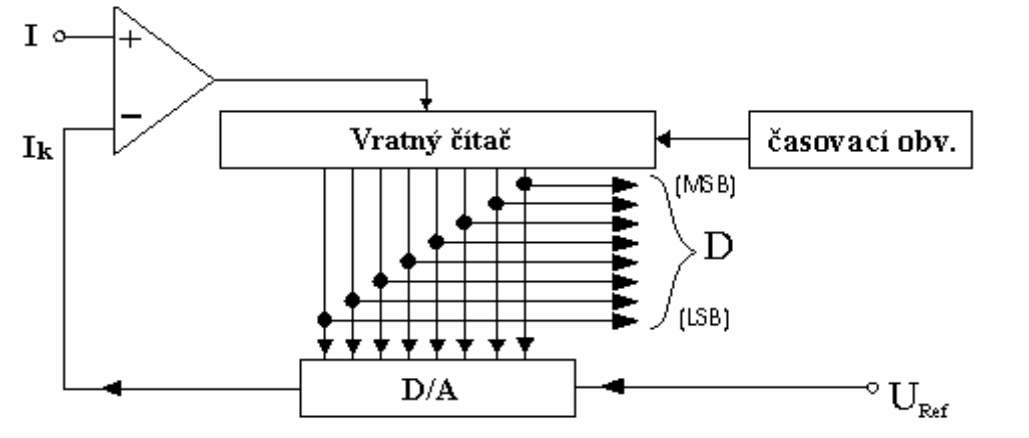
Metody vícesklonných integrací používáme pro redukci multiplikativních a aditivních chyb převodu. U třítaktní integrace je první takt vyhrazen pro korekci aditivních chyb. Čtyřtaktní integrace má na korekci vyhrazeny první dva takty. Další dva takty jsou už měřicí.

1.3.6 Kompenzační (sledovací) převodníky

Tyto převodníky měří okamžitou hodnotu měřeného signálu a jsou opět principielně jednoduché. Jsou schématicky podobné aproximačním, ale místo aproximačního registru mají vratný čítač, viz obr. 14. Obsahují D/A převodník generující schodovitý proud I_k, který je čítán čítačem až do doby, než přeroste přes měřený proud I. Až se tak stane, tak vstupní komparátor změní stav a způsobí, že čítač začne čítat vzad. Pokud tedy napětí na vstupu rovnoměrně poroste, budou generovány „normální schody“. Kdybychom na vstup přiváděli konstantní stejnosměrné napětí, tak by komparátor stále měnil svůj stav, vratný čítač by se stejnou frekvencí čítal vpřed i vzad a bylo by generováno napětí o obdélníkovém průběhu a se stálou efektivní hodnotou. Mezní měřitelný kmitočet lze pro sinusový průběh proudu získat ze vztahu (7).

$$f_m \geq \frac{2^{-n}}{\pi} \cdot f_g , \tag{7}$$

kde f_g je kmitočet generátoru převodníku, f_m je max. měřitelný, n je počet bitů ADC



Obr. 14 Kompenzační převodník

1.3.7 Převodníky se sigma delta modulací

Jejich princip je ve vyrovnávání náboje se vzorkovanou zpětnou vazbou, která udržuje nulovou střední hodnotu náboje na integračním kondenzátoru. Tyto převodníky jsou tvořeny sigma delta modulátorem, který pomocí referenčního napětí obou polarit, převede průběh měřeného napětí na pilový a tento průběh je vzorkován vysokou frekvencí pomocí klopného obvodu typu D. Výstup klopného obvodu má pak pulsní průběh s proměnnou střídou a je synchronizovaný vzorkovacím kmitočtem. Dále je signál veden do číslicového filtru a decimátoru. V těchto obvodech je signál kmitočtově redukován. Původní analogový signál je teď zakódován, pomocí referenčního napětí a dob trvání výstupních měřených pulsů. Tyto převodníky mají velmi vysoké rozlišení, a to až 24 bitů. Jsou vhodné pro zvukové signály. Převzorkováním zvýšíme odstup S/N na hodnotu:

$$S/N = 6{,}02 \cdot n + 1{,}76 + 10 \log K\,, \tag{8}$$

kde K je násobek původního vzorkovacího kmitočtu a n počet bitů ADC.

1.3.8 Ostatní A/D převodníky

- **Převodníky s vyrovnávací strukturou**
- **Převodníky s technologií MagAmp**
- **Interpolační převodníky**
- **Polohové optické převodníky**

2. Popis použitých obvodů

2.1 AD7675 – A/D převodník

Tento analogově-digitální převodník od společnosti ANALOG DEVICES vyhovuje téměř všem požadavkům, proto budou dále popsány jeho základní parametry a důležité informace pro jeho správné použití.

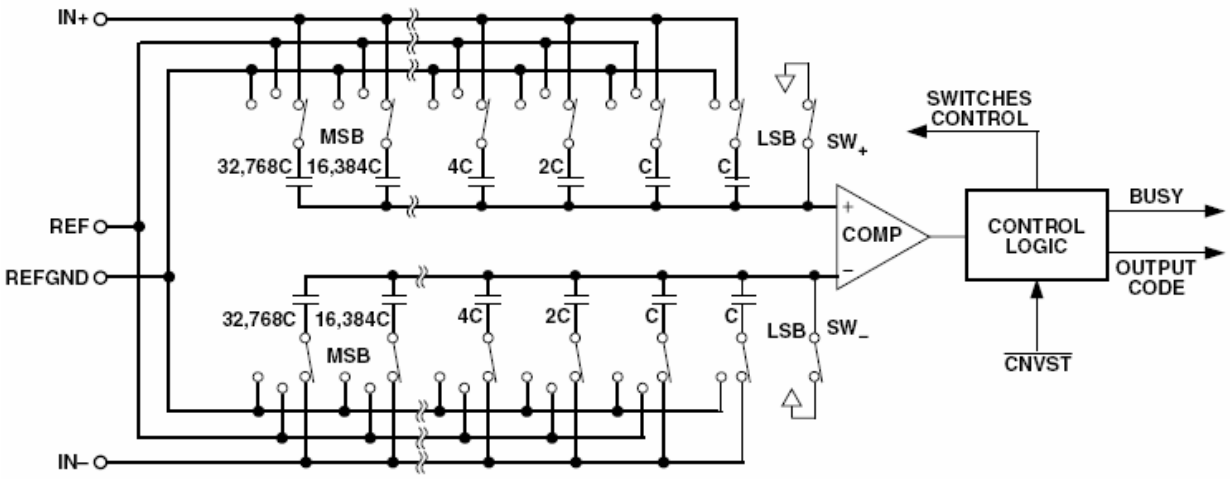
2.1.1 Vlastnosti obvodu AD7675

Jedná se o jednokanálový šestnáctibitový analogově-digitální převodník poskytující všechny požadované vlastnosti. Podrobné informace naleznete především v [12].

Základní princip funkce je systém postupné aproximace s přerozdělováním náboje. Jedná se však o vylepšenou metodu označovanou jako PulSAR®. (Success Aproximation Register, Pul nemá žádný význam, důvod zavedení je kvůli podobnosti celého názvu s vesmírnými objekty). Vzorkovací rychlost je 100 kSPS (100 kHz). Celkové harmonické zkreslení (THD) se pohybuje okolo -110 dB. Integrální nelinearita (INL) je maximálně 1,5 LSB. Odstup signál/(šum+zkreslení) (S/(N+D)) je 94 dB. Vstupní rozsah měřitelného napětí je ± 2,5 V, diferenciálně mezi dvěma vstupy. Vstupní napětí tedy není měřeno mezi vstupem a zemí, nýbrž mezi dvěma vstupními piny. Napájecí napětí musí být v mezích 5 ± 0,25 V. Organizaci výstupních dat si můžeme vybrat mezi paralelní nebo sériovou. Výstup může být klasický binární nebo ve dvojkovém doplňku. Při paralelním výstupu dat si lze vybrat ze dvou možností, na kterých určených osmicích pinů bude k dispozici horní bajt, a na kterých bajt dolní. Při použití sériového výstupu dat je k dispozici 3V nebo v 5V logika výstupu.

Je vyroben 0,6μm technologií CMOS. Má interní generátor hodinového kmitočtu realizovaný pomocí krystalu. Napěťová reference 2,5 V se připojuje externě. Pro zlepšení přesnosti převodu analogového napětí na digitální data jsou v převodníku vestavěny obvody pro detekci a korekci chyb. Příkon tohoto převodníku se pohybuje okolo 15 mW. Je vyráběn pouze v provedení SMD (Surface Mount Device), a to ve dvou konstrukčních variantách. První variantou je 48-pinové LFCSP (Lead Frame Chip Scale Package), jenž je navrženo pro speciální patici. V našem případě byla použita druhá varianta, tedy 48-pinové LQFP (Lead Quad Flatpack Package). Rozměry obou pouzder jsou téměř shodné. Pouzdro má čtvercový půdorys o straně 7 mm, přičemž na každé straně je 12 vývodů. Z tohoto vyplývá, že jednotlivé vývody i mezery mezi nimi mají šířku jen několik desetin milimetru.

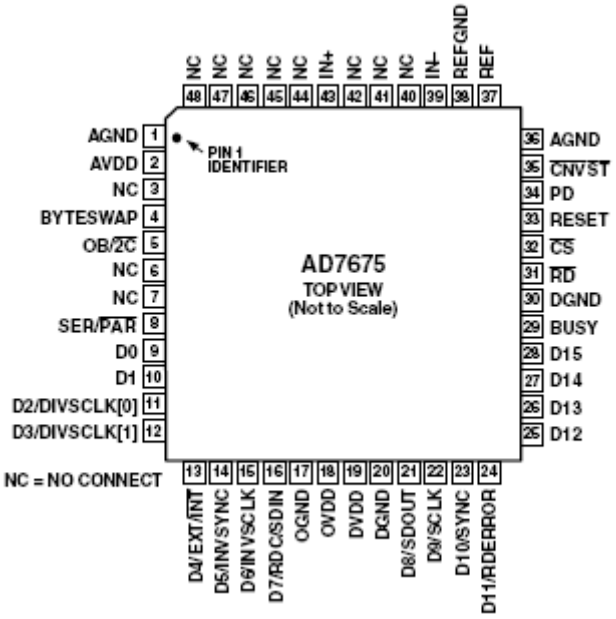
Na obr. 15 je funkční schéma tohoto A/D převodníku. Jde o systém postupné aproximace s přerozdělováním náboje, který byl popsán v kapitole o typech A/D převodníků.



Obr. 15 Funkční schéma AD7675

2.1.2 Popis jednotlivých vývodů

Ve vyhotoveném zařízení je použito 48-pinové pouzdro LQFP, viz obr. 16. Bez základní představy o funkci jednotlivých vývodů, nemusí být dostatečně zřejmý princip funkce popisovaného zařízení. Na následujících stránkách bude proto popsán hlavní účel všech vývodů, včetně možností jejich využití.



Obr. 16 Rozmístění vývodů na pouzdře LQFP

POPIS VÝVODŮ

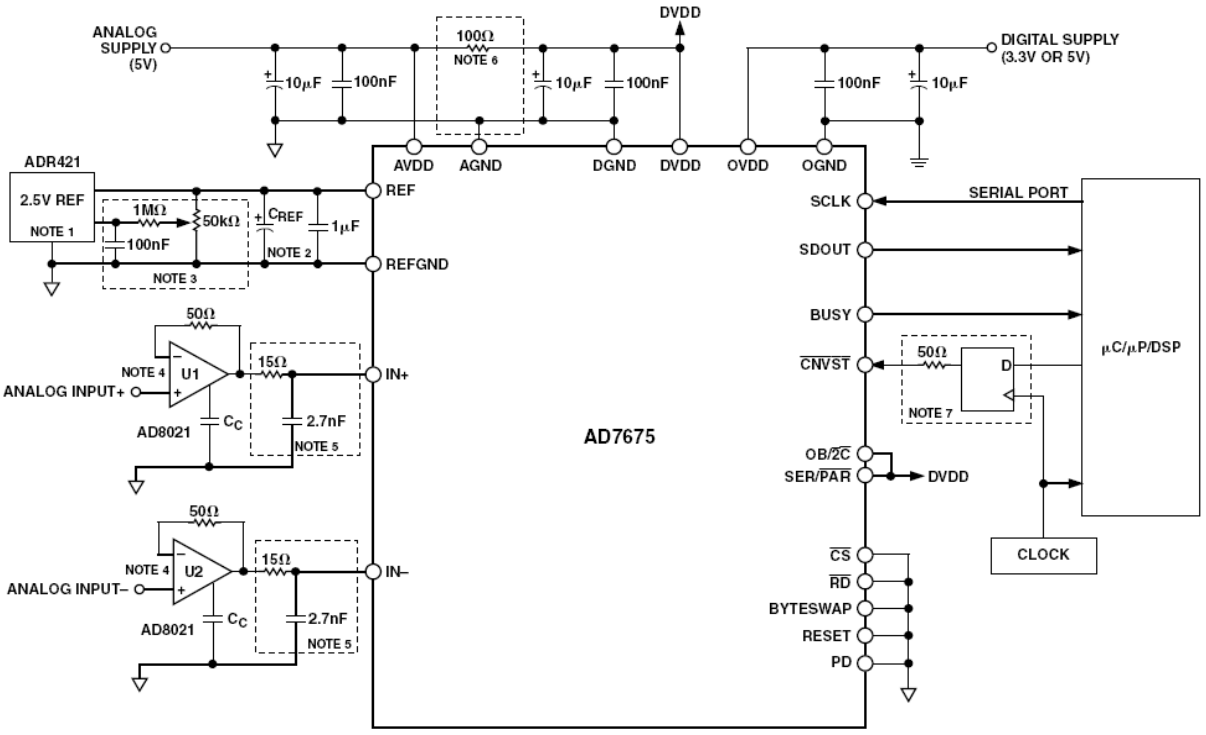
1. **AGND** – Analogová napájecí zem
2. **AVDD** – Analogové napájení +5 V
3. , 6. , 7. , 40.-42. , 44.-48. **NC** – Nezapojeno
4. **BYTESWAP** – Přepínání umístění výstupních dat v paralelním režimu. Pokud je na něj přivedená logická nula, pak dolní bajt výsledku je vyveden na pinech 0-7, horní bajt pak na pinech 8-15. Logickou jedničkou je poloha obou bajtů výsledku přehozena.
5. **OB/2C** – Volba kódování výstupních dat. Přivedením logické nuly jsou výstupní data kódována ve dvojkovém doplňku. Logická nula způsobí, že výstup dat je ve standardním binárním režimu
8. **SER/PAR** – Volba režimu výstupních dat. Pokud na tento pin přivedeme logickou nulu tak je zvolen paralelní výstup dat. Logickou jedničku přivedeme, pokud si přejeme sériový výstup
9. , 10. **DATA[0:1]** – Bity 0 a 1 paralelního výstupu dat. Je-li zvolen sériový režim, pak jsou ve stavu vysoké impedance
11. , 12. **DATA[2:3]/ DIVSCLK[0:1]** – Bity 2 a 3 paralelního výstupu dat. Je-li zvolen sériový režim, pak slouží, podle dalších nastavení, pro řízení po sériové lince.
13. **DATA[4]/EXT/INT** – Bit 4 paralelního výstupu dat. Je-li zvolen sériový režim, pak slouží pro volbu externího nebo interního zdroje hodin při sériové komunikaci
14. **DATA[5]/ INVSYNC** – Bit 5 paralelního výstupu dat. Je-li zvolen sériový režim, pak slouží pro volbu aktivního stavu synchronizačního signálu, při sériové komunikaci
15. **DATA[6]/INVSCLK** – Bit 6 paralelního výstupu dat. Je-li zvolen sériový režim, pak slouží pro invertování řídícího hodinového signálu
16. **DATA[7]/RDC/SDIN** – Bit 7 paralelního výstupu dat. Je-li zvolen sériový režim, pak slouží pro jako externí datový vstup nebo pro aktivaci čtecího režimu, dle pinu 13
17. **OGND** – Digitálního napájecí zem při spojení periferií
18. **OVDD** – Digitální napájení +5 V nebo +3 V, podle způsobu spojení s ostatními periferiemi, v režimu master je tento pin výstupním, jinak je vstupní
19. **DVDD** – Digitální napájení +5 V
20. **DGND** – Digitálního napájecí zem
21. **DATA[8]/SDOUT** – Bit 8 paralelního výstupu dat. Je-li zvolen sériový režim, pak slouží jako výstup dat, synchronizovaný hodinovým signálem na pinu 22
22. **DATA[9]/SCLK** – Bit 9 paralelního výstupu dat. Je-li zvolen sériový režim, pak slouží pro jako vstup nebo výstup synchronizačního signálu, podle stavu pinu 13

23. **DATA[10]/SYNC** – Bit 10 paralelního výstupu dat. Je-li zvolen sériový režim, pak slouží pro jako výstup synchronizačního rámce
24. **DATA[11]/RDERROR** – Bit 11 paralelního výstupu dat. Je-li zvolen sériový režim, tak se tento bit stane příznakovým, indikujícím nekompletní přečtení nebo vyslání dat
25.-28. **DATA[12:15]** – Bity 12 až 15 paralelního výstupu dat
29. **BUSY** – Výstup, indikující zaneprázdnění A/D převodníku. Pokud je nastartována konverze, pak je tento pin v logické jedničce, a to až do ukončení konverze
30. **DGND** – Digitálního napájecí zem
31. $\overline{RD}$ – Přečtení dat. Pro povolení čtení výstupních dat, musí být CS i RD na logické nule
32. $\overline{CS}$ – Zvolení čipu. Spolu s RD musí být na logické nule, aby zapojení plnilo svou funkci
33. **RESET** – Resetovaní vstup. Logická jenička způsobí resetování převodníku
34. **PD** – Zapínání úsporného režimu. Přivedená logická jednička aktivuje úsporný režim
35. $\overline{CNVST}$ – Start konverze měřeného napětí. Sestupná hrana na tomto vstupu způsobí start převodu. Pokud je pin stále na úrovni logické nuly, převod je neustále startován. Je-li pin v logické jedničce, převod neprobíhá
36. **AGND** – Analogová napájecí zem
37. **REF** – Vstup napěťové reference
38. **REFGND** – Zem napěťové reference
39. **IN–** – Záporný analogový vstup
43. **IN+** – Kladný analogový vstup

2.1.3 Doporučené zapojení

Při návrhu vlastního zapojení se vycházelo především z doporučených zapojení jednotlivých obvodů. To platí i pro zapojení tohoto A/D převodníku, jehož doporučené zapojení je na obr. 17. Jako napěťovou referenci však nebyl použit doporučený obvod ADR421, nýbrž TL431 se stejným výstupním napětím, ale s nutností připojení mnohem menšího počtu externích součástek a snazším funkčním zapojením. Jelikož digitální napájení není využito, jsou jeho vstupní piny připojeny k napájení analogovému přes jednoduchý RC filtr. Na doporučeném zapojení je použito sériového režimu výstupu dat. V našem případě je ale požadován režim paralelní. Navíc nebylo možné použít krystal (16 MHz) mikrokontroléru k řízení spouštění převodu na pinu CNVST (100 kHz). Proto je spojení A/D převodníku s mikrokontrolérem naprosto odlišné. Na vstupech nejsou zapojeny operační zesilovače, nýbrž pouze dvě Zenerovy diody sloužící jako přepěťová ochrana.

Na vstupu je také zařazen RC filtr typu dolní propust. V doporučeném zapojení jsou použité takové hodnoty součástek filtru, že pokles o 3 dB nastává až na kmitočtu přibližně asi 2,5 MHz, což je zbytečně mnoho. Proto byly součástky zvoleny tak, aby horní mezní kmitočet byl přibližně 105 kHz. Jelikož vzorkovací kmitočet A/D převodníku je 100 kHz, můžeme tedy bez velkých chyb vzorkovat signály s frekvencí maximálně asi 20 kHz.



Obr. 17 Doporučené zapojení AD7675

2.2 FT232BM – převodník UART/USB

Jedná se o oboustranný USB/UART převodník od firmy FTDI, viz [13]. Jednotka UART je využívána pro komunikaci, při které nám postačí jen dva datové vodiče. Použití tohoto obvodu je nezbytné, protože jednotka UART v mikrokontroléru pracuje s jinými logickými úrovněmi než sběrnice USB v počítači. Verze FT232BL je funkčně naprosto stejná, je však vyrobena s téměř nulovým obsahem olova.

2.2.1 Vlastnosti obvodu FT232BM

K tomuto obvodu lze snadno připojit paměť typu E^2PROM, která slouží pro uchování identifikačních údajů a dalších nastavení. Paměť lze naprogramovat přímo z počítače přes USB. Obvod lze přes USB také napájet, a pomocí speciálního výstupu lze toto napájecí napětí postoupit dalším možným obvodům. Hlavní integrované součásti jsou: regulátor napětí pro USB, násobička frekvence z 6 MHz na 48 MHz a vlastní konvertor úrovní UART (pro logiku 3,3 V i 5 V). Obvod podporuje veškeré služby nutné pro sériovou komunikaci. Při speciálním použití je podporován také přenos s rychlostí až 3 Mb/s pro TTL logiku. Obvod má zabudovanou podporu událostních znaků přerušení linky, na přijímači lze nastavit time-out. Hloubka přijímacího bufferu je 384 bajtů a vysílacího 128 bajtů. Je standardně vyráběn ve dvou variantách 32-pinových SMD pouzder, a to LQFP (7x7 mm) nebo QFP (5 x 5 mm).

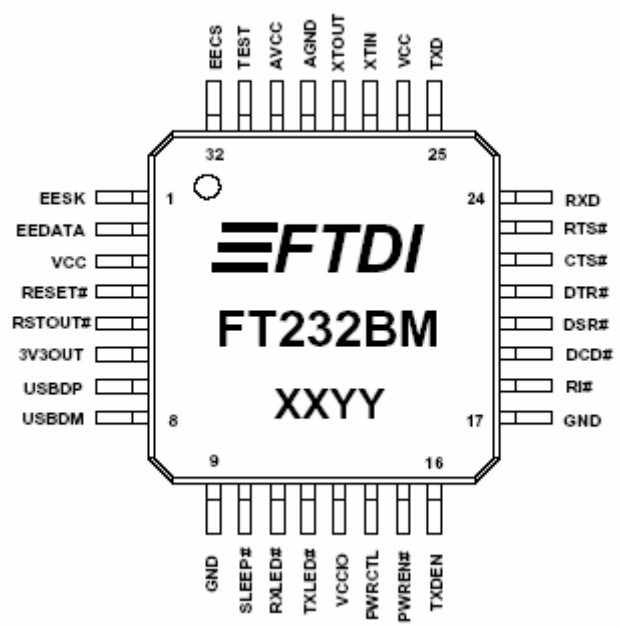
Vlastní softwarové ovládání obvodu se provádí pomocí ovladačů VCP či D2XX. Ovladač VCP si USB port představuje jako normální sériový. D2XX poskytuje více možností, a proto je výhodnější použít tento ovladač. S obvodem FT232BM komunikuje uživatelský program přes dynamickou knihovnu FTD2XX.DLL a přes ovladač FTD2XX.SYS, ale také přes speciální zásobník USB pro Windows. Ovládání obvodu speciálním softwarem umožní načíst do paměti E^2PROM potřebné informace. Před začátkem jakékoli operace je však nutné získat tzv. Handle obvodu pomocí speciální funkce. Je-li Handle získán, jsme oprávněni k provádění výše zmíněných operací.

Obvod dosahuje při napájení 5 V maximální výkonové ztráty 500 mW. Napájecí napětí VCC se může pohybovat od -0,5 V do 6 V. Výstupní proud z UART nesmí překročit 24 mA, její vstupní napětí musí být nižší než VCC + 0,5 V. Je nutné externě připojit oscilátor. Pro zdroj kmitočtu 6 MHz lze použít keramický rezonátor nebo přímo krystal. Vhodné třívývodové rezonátory mají blokovací kondenzátory integrované, proto se nemusí žádné další připojovat. Při připojení krystalu se použijí blokovací kondenzátory s kapacitou 27 pF.

Aby vnitřní UART pracovalo v 3,3V logice, musí být na VCCIO napětí 3,3 V, jež můžeme například získat pomocí nízkopříkonového snižujícího regulátoru z původního nap. napětí z USB. Pro napájení logiky 3,3 V lze využít výstup 3V3OUT, ale proudový odběr musí být do 5 mA.

Obvod disponuje speciálním výstupním režimem – Bit Bang. V tomto režimu lze přímo řídit všech 8 datových linek UART (RI#, DCD#, DSR#, DTR#, CTS#, RTS#, RXD, TXD). Tyto linky tedy dohromady tvoří univerzální I/O sběrnici širokou 8b. Data vstupující do obvodu jsou sekvenčně posílána na toto rozhraní, a to s přenosovou rychlostí nastavenou pomocí předděličky. Rychlost přenosu bajtů z výstupního bufferu je stejná jako vzorkovací frekvence, se kterou se bajty ukládají do bufferu vstupního. Přesné zapojení jednotlivých vývodů je na obr. 18.

2.2.2 Popis jednotlivých vývodů



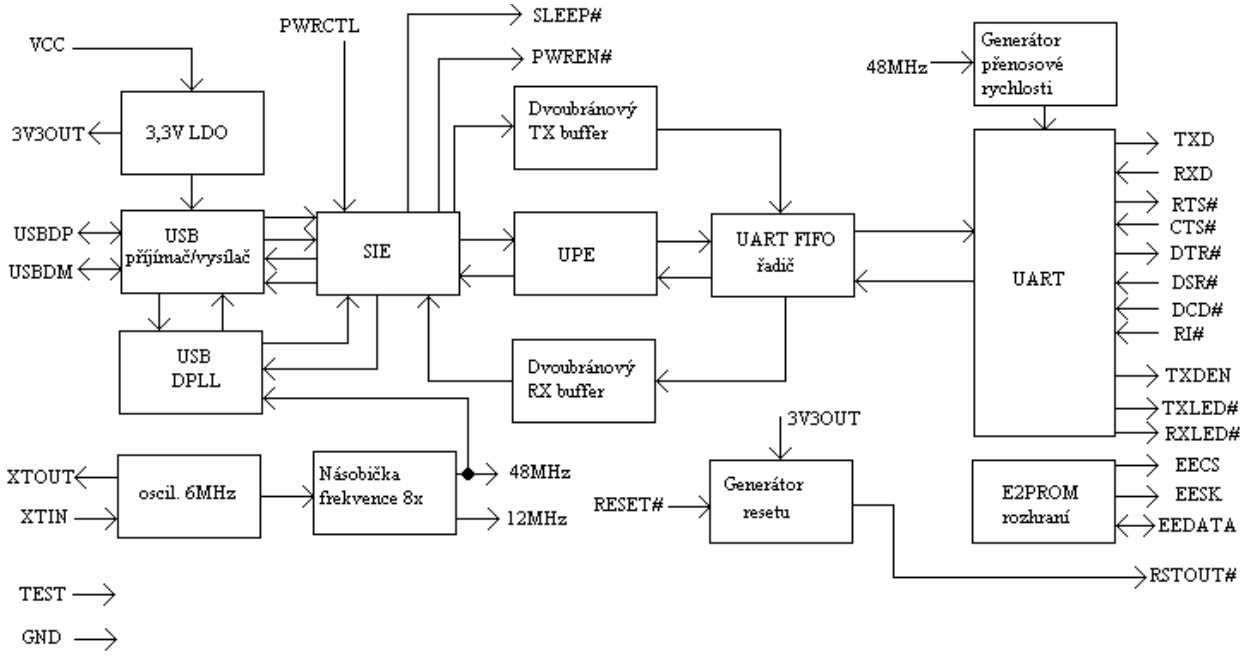
Obr. 18 Pouzdro LQFP-32 s vývody obvodu FT232BM

XX je označení roku výroby, YY je pořadové číslo měsíce výroby

POPIS VÝVODŮ

1. **EESK** – Výstup hodinového signálu pro paměť EEPROM
2. **EEDATA** – I/O datový pin, je nutný sériově připojený rezistor 2k2 a 10k pull-up rezistor
3. **VCC** – Napájecí vstup
4. **RESET**# – Vstup pro resetování nadřazeným obvodem, jinak připojeno na VCC
5. **RSTOUT**# – Resetovací vstup, je-li pin 4 používán, pak je ve stavu vysoké impedance
6. **3V3OUT** – Výstup integrovaného regulátoru, použití i pro vnější napájení do 5 mA
7. **USBDP** – I/O USB datový pin pozitivní
8. **USBDM** – I/O USB datový pin negativní
9. **GND** – Zem pro napájení a data z jednotky UART
10. **SLEEP#** – Výstup, při úsporném režimu přejde do log. 0, slouží jako obecný vypínač
11. **RXLED**# – Při příjmu dat na USB vytvoří puls do log. 0, (indikace pomocí LED)
12. **TXLED**# – Při vyslání dat na USB vytvoří puls do log. 0, (indikace pomocí LED)
13. **VCCIO** – Napájení pro UART rozhraní 3 - 5,25 V. Připojit na VCC pro 5V logiku
14. **PWRCTL**# – Vstup, pro napájení z USB připojit na zem, při vnějším napájení na VCC
15. **PWREN**# – Výstup, v úsporném režimu je v log. 1. Řízení napájení dalších obvodů
16. **TXDEN** – Aktivace vysílání dat pro sběrnici RS485
17. **GND** – Zem pro napájení a data z jednotky UART
18.-23. **RI#, DCD#, DSR#, DTR#, CTS#, RTS#** – Signály modemu
24. **RXD** – Vstup přijímaných dat, např. z UART
25. **TXD** – Výstup vysílaných dat, např. z UART
26. **VCC** – Vstup napájení jádra 5 V
27. **XTIN** – Vstup oscilátoru 6 MHz, nebo vstup vnějšího hodinového signálu
28. **XTOUT** – Výstup kmitočtu 6 MHz, např. pro vnější obvody
29. **AGND** – Zem zabudované násobičky hodin
30. **AVCC** – Napájení zabudované násobičky hodin, 5 V
31. **TEST** – Vstup, je-li v log. 1, pak jde zapnut normální režim, log.0 – testovací režim
32. **EECS** – Tzv. Chip Select paměti EEPROM, nutný pro její spolupráci s FT232BM

2.2.3 Blokové schéma



Obr. 19 Blokové schéma FT232BM

HLAVNÍ ČÁSTI PŘEVODNÍKU

3,3 Low Drop Out regulátor

Zde se generuje napětí 3,3 V pro buzení USB vysílače a pro výstup RSTOUT#. Napájení vnějších zařízeních z tohoto zdroje je omezeno maximálním odběrem 5 mA.

USB přijímač/vysílač

Vlastní rozhraní pro připojení kabelu s USB konektorem.

USB DPLL

Jedná se o detekci datového a hodinového signálu z NRZI kódování USB. NRZI je speciální způsob kódování, kdy se hodinový signál získává z datového tak, že se nula v datech bere jako impuls pro změnu stavu takto tvořeného hodinového signálu.V případě jedničky je signál konstantní, ať už zůstal na úrovni logická nula či jednička hodinového pulsu.

Násobička kmitočtu

Hodinový signál je násoben dvěma a osmi. Dvojnásobný hodinový kmitočet jde do SIE, UPE A UART. Osminásobný je potřebný pro funkci USB DPLL.

SIE – Serial Interface Engine

Podle standardu USB 1.1 vkládá a vyjímá synchronizační bity z přenosu.

UPE – USB protocol engine

Spravuje datový proud z řídícího koncového bodu USB.

Dvoubránový TX buffer

Do bufferu se ukládají data z výstupního koncového bodu USB. Data jsou následně odebírána vysílacím registrem UART. Obě operace s daty jsou řízeny UART FIFO řadičem.

Dvoubránový RX buffer

Do bufferu se ukládají data ze vstupního přijímacího registru UART, aby mohla být posléze zase odebrána do obvodu SIE. Pro odstranění zbytků dat z tohoto bufferu se používá tzv. TIME-OUT, který je navíc programovatelný, a to nám navíc umožní přesně nastavit požadovaný okamžik jeho aktivace.

UART FIFO řadič

Slouží jako řídící jednotka přenosu mezi oběma typy bufferů a I/O registrem UART.

UART

Je to registr, který provádí sériově-paralelní a paralelně-sériovou konverzi dat na rozhraní RS232. Signál TXDEN je aktivační signál vysílače pro ovládání RS485 vysílačů.

Generátor přenosové rychlosti

Přenosovou rychlost lze pomocí generátoru naprogramovat od 300 b/s do 3 Mb/s. Generátor představuje 14 b dělička a 3bitový registr pro zjemnění rozsahu nastavení rychlosti přenosu.

Generátor resetu (Power On Reset)

Slouží pro resetování vlastního obvodu nebo pro vnější periferie. Resetování se děje pomocí pinů RESET#, resp. RSTOUT#. Vstup RESET# se však doporučuje připojit na VCC.

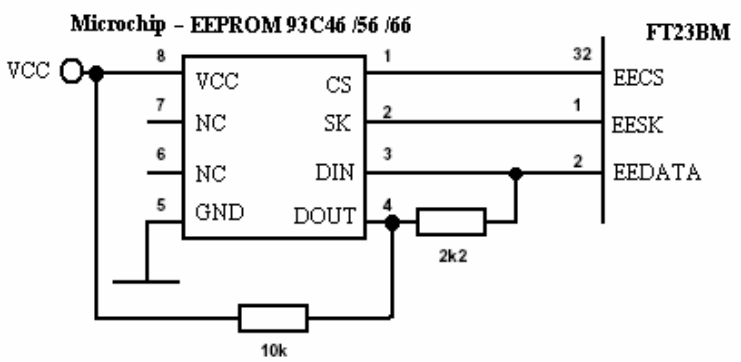
E2PROM rozhraní

Paměť E^2PROM slouží k uložení dat hlavně pro OEM aplikace, např. popis výstupu, informace o odběru proudu, sériové číslo, Product Descriptor. Je také vyžadována, pokud je v měřícím obvodu počítače připojeno více obvodů FT232BM. Je-li paměť prázdná či jen nepřipojena jsou použity interní přednastavené hodnoty o odběru proudu apod. Výhodou paměti je snadná programovatelnost přímo na desce přes USB. V této paměti se zapíná izochronní přenosový režim. Izochronní přenos dat je jeden ze čtyř druhů přenosů po USB.

2.2.4 Doporučená zapojení

Připojení paměti EEPROM k FT232BM

Z obr. 20 je patrné, že vývody EECS, ESK a EEDATA jsou k paměti připojeny přímo, bez jakýchkoliv součástek. Rezistor 2k2 mezi vývody DOUT a DIN slouží k tomu, aby mezi sebou nekolidovala vstupní a výstupní data, která jsou do FT232BM vedena po jednom vodiči. Při připojení napájení nebo při resetu z počítače se provede její kontrola a načtení dat.

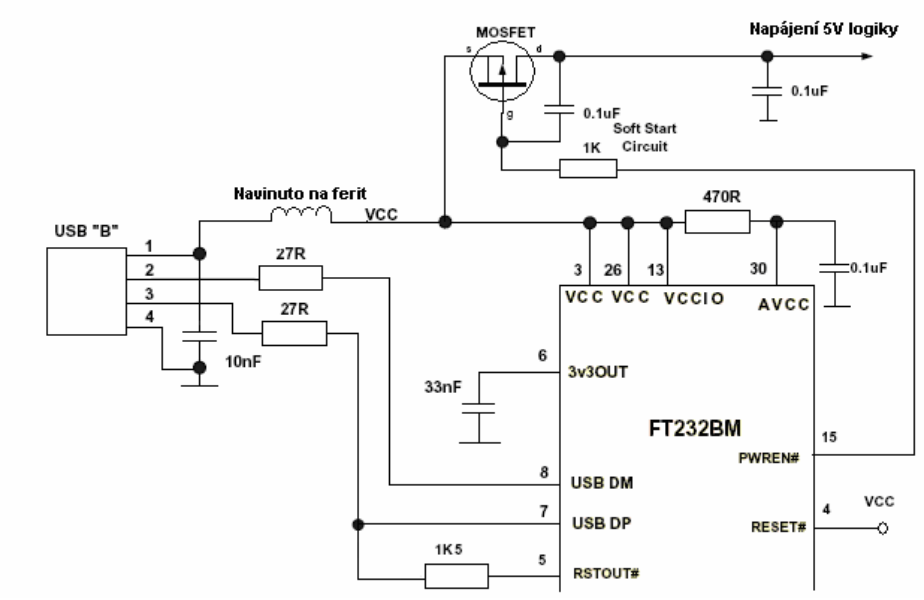


Obr. 20 Připojení sériové paměti E^2PROM

Napájení z USB a řízení napájení pro další obvody

Jak již bylo napsáno, odebíraný proud z USB sběrnice je okamžitě po připojení omezen 100 mA nebo 500 μA v režimu USB suspend. Chceme-li řídit připojení napájení pro zařízení s odběrem nižším než 100 mA, pak můžeme použít vývod PWREN#, který po úspěšné enumeraci, tedy po úspěšném rozpoznání připojeného zařízení, připojí napájecí napětí do ostatních obvodů. Ze sběrnice USB nemůžeme celkově odebírat více než 500 mA. Aby byl deskriptor (informace o zařízení) z paměti EEPROM brán v úvahu, je nutné vývod PWRCTL# připojit na log. 0. Pro lepší stabilitu obvodu se mezi VCC a GND připojí blokovací kondenzátory 10 μF a 100 nF. Tlumivka na vstupu slouží pro omezení rušení. Zapojení je možné modifikovat také tak, že cívku nezařadíme. Záleží na okolních podmínkách. Pro řízení napájení dalších obvodů připojených na USB se použije MOSFET tranzistor, podle zapojení na obr. 21.

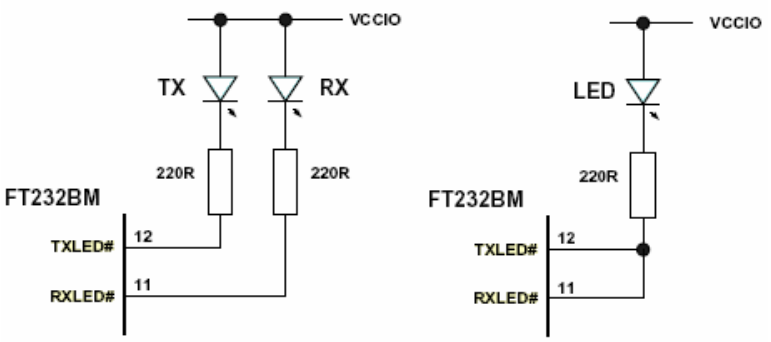
Pokud je odběr dalších zařízeních vyšší než 100 mA, tak je vhodnější použít obvod s měkkým startem. Při použití tohoto zapojení je však nutné nastavit v E^2PROM soft pull-up. Pracujeme-li s 3,3V logikou, pak je vývod PWREN# připojen na 3V3OUT a VCCIO je napájen z jiného zdroje 3,3 V a nelze tedy současně odpojit logiku i vstup VCCIO. Pokud je toto současné vypnutí žádoucí, lze VCCIO také připojit na 3V3OUT.



Obr. 21 Napájení z USB a řízení napájení periferií tranzistorem

Způsob indikace příjmu či vysílání dat

Indikace je realizovaná pomocí dvou výstupů pro LED diody. Příjem indikuje LED na výstupu RXLED#, vysílání zase TXLED#. Spínání se provádí automatickým uzemněním těchto výstupů. Aby při rychlé komunikaci bylo možné postřehnout blikání diod, je použit vnitřní monostabilní klopný obvod. Oba vývody lze spojit a tak lze sledovat pouze činnost či nečinnost zařízení při žádaném transferu dat. Zapojení diod je na obr. 22.



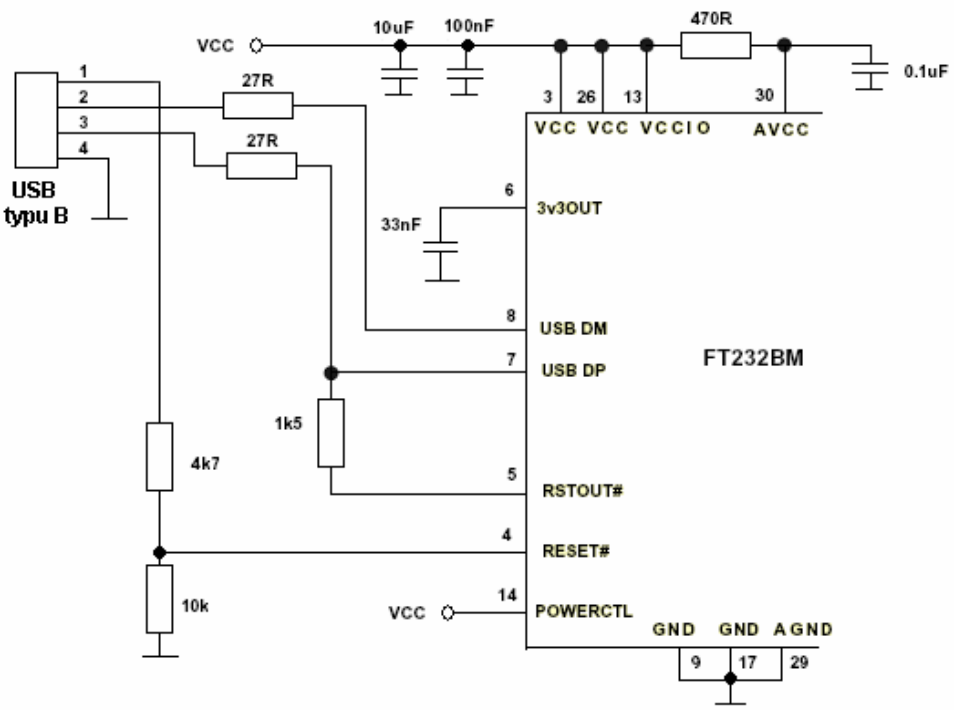
Obr. 22 Možnosti sledování transferu dat

Napájení z vlastního zdroje

Pro tuto situaci je více podobných řešení. Základní zapojení je na obr. 23. Po odpojení napájení z USB je FT232BM ve stavu resetu, ale oscilátor stále pracuje a tudíž způsobuje zbytečný proudový odběr. Tato nepříjemnost se dá odstranit jistou modifikací tohoto zapojení, ve které přejde FT232BM díky tranzistoru velice rychle do spánkového režimu a vstup RESET# zůstává volný.

Obvod takto napájený má první výhodu v možnosti většího proudového odběru zařízení. Druhou důležitou výhodou je to, že se nám do připojeného zařízení neindukuje rušení z počítače.

Mezi RSTOUT# a USBDP je trvale připojen zdvihací rezistor. Resetování je řízeno pomocí děliče napětí na napájecím výstupu z USB. Při resetu jsou vývody UART v tzv. třetím stavu, ale jelikož mají integrovány zdvihací rezistory, jsou taženy spíše k log.1.



Obr. 23 Napájení z vlastního zdroje

2.3 ATmega16 – mikrokontrolér

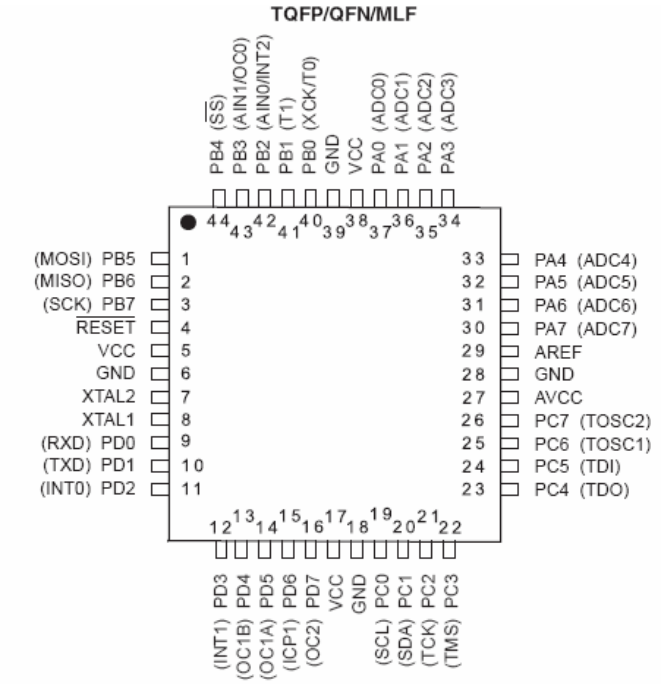
Jedná se o výkonný osmibitový CMOS mikrokontrolér AVR od firmy ATMEL. Je podporován mnoha balíky programů a aplikačních zařízení. Má mnoho funkcí, z nichž ty nejzajímavější budou dále stručně popsány. Podrobnější informace lze nalézt v [11].

2.3.1 Vlastnosti obvodu ATmega16

Mikrokontrolér využívá pokročilou architekturu RISC, a to s rychlostí až 16 MIPS při 16 MHZ. Jedná se o neakumulátorový typ obsahující 32 všeobecně použitelných osmibitových registrů. Spojení mikrokontroléru s okolím je zajištěno čtyřmi osmipinovými vstupně/výstupními porty. Instrukční soubor obsahuje 131 instrukcí. Používá Harvardskou koncepci rozdělení paměti, tedy oddělenou pro data a pro program. Programová paměť typu FLASH má kapacitu 16 kB s možností až 10 000 přepisů. Dále obsahuje 1 kB datové paměti SRAM a 512 B paměti E^2PROM.

Mikrokontrolér obsahuje dva integrované osmibitové čítače/časovače a jeden šestnáctibitový se záchytným módem, dále analogový komparátor, programovatelné sériové rozhraní USART a čtyři PWM kanály. Potom je zde ještě čítač reálného času s odděleným oscilátorem a desetibitový A/D převodník s osmi vstupními kanály, přičemž je možné využit speciálního režimu pro snížení šumu. Ke speciálnímu vybavení patří např. obvod Power-on Reset, kalibrovaný RC oscilátor, obvod pro vnitřní i vnější odpojení zdrojů a šest módů spánku. Napájení musí být mezi 4,5 – 5,5 V nebo u verze „L“ 2,7 – 5,5 V. Napájení mikrokontroléru je nutné blokovat kondenzátory umístěnými co nejblíže k tomuto obvodu. Prostor pod mikrokontrolérem je doporučeno nechat vyplněný mědí a tuto plochu pak spojit s příslušnou napájecí zemí. Vyrábí se v několika variantách pouzder, a to DIP 40 nebo dvě varianty SMD. Pouzdro pro SMD montáž má oproti DIP o čtyři piny více. Tyto redundantní vývody jsou však pouze multiplikací napájecích pinů. Navíc ani číslování vývodů není u DIP a SMD verze pouzdra stejné. Jeden typ vyráběného pouzdra je na obr. 24, tento typ byl také použit pro popisované zařízení.

2.3.2 Popis jednotlivých vývodů



Obr. 24 Pouzdro TQFP 44

POPIS VÝVODŮ

1. **PB5/MOSI** – Obecný pin portu B nebo komunikační pin rozhraní SPI
2. **PB6/MISO** – Obecný pin portu B nebo komunikační pin rozhraní SPI
3. **PB7/SCK** – Obecný pin portu B nebo synchronizační pin rozhraní SPI
4. $\overline{RESET}$ – Resetovací vstup, resetovací signál je puls log. 0, delší než 50 ns
5. **VCC** – Vstup napájení +5 V
6. **GND** – Zem pro napájecí napětí
7. **XTAL2** – Výstup invertujícího zesilovače zabudovaného oscilátoru
8. **XTAL1** – Vstup do invertujícího zesilovače oscilátoru a vstup hodin řídícího okruhu
9. **PD0/RXD** – Obecný pin portu D nebo vstup dat jednotky USART
10. **PD1/TXD** – Obecný pin portu D nebo výstup dat jednotky USART
11. **PD2/INT0** – Obecný pin portu D nebo vstup vnějšího přerušení 0
12. **PD3/INT1** – Obecný pin portu D nebo vstup vnějšího přerušení 1
14. **PD5/OC1A** – Obecný pin portu D nebo výstupní pin Compare čítače/časovače 1A

15. **PD6 /ICP** – Obecný pin portu D nebo vstupní pin Capture čítače/časovače 1
16. **PD7/OC2** – Obecný pin portu D nebo výstupní pin Compare čítače/časovače 2
17. **VCC** – Vstup napájení +5 V
18. **GND** – Zem pro napájecí napětí
19. **PC0/SCL** – Obecný pin portu C nebo synchronizační signál pro rozhraní TWI
20. **PC1/SDA** – Obecný pin portu C nebo datový pin pro rozhraní TWI
21. **PC2/TCK** – Obecný pin portu C nebo synchronizační signál JTAG rozhranní
22. **PC3/TMS** – Obecný pin portu C nebo výběr režimu JTAG rozhraní
23. **PC4/TDO** – Obecný pin portu C nebo výstup dat z JTAG rozhraní
24. **PC5/TDI** – Obecný pin portu C nebo vstup dat pro JTAG rozhraní
25. **PC6/TOSC1** – Obecný pin portu C nebo první pin časování vnějšího oscilátoru
26. **PC7/TOSC2** – Obecný pin portu C nebo druhý pin časování vnějšího oscilátoru
27. **AVCC** – Napájecí napětí +5 V pro vnitřní A/D převodník
28. **AGND** – Zem pro referenční napětí
29. **AREF** – Vstup referenčního napětí, standardně 2,5 V
30. **PA7**/**ADC7** – Obecný pin portu A nebo vstup 7. kanálu A/D převodníku
31. **PA6**/**ADC6** – Obecný pin portu A nebo vstup 6. kanálu A/D převodníku
32. **PA5**/A**DC5** – Obecný pin portu A nebo vstup 5. kanálu A/D převodníku
33. **PA4**/**ADC4** – Obecný pin portu A nebo vstup 4. kanálu A/D převodníku
34. **PA3**/**ADC3** – Obecný pin portu A nebo vstup 3. kanálu A/D převodníku
35. **PA2**/**ADC2** – Obecný pin portu A nebo vstup 2. kanálu A/D převodníku
36. **PA1**/**ADC1** – Obecný pin portu A nebo vstup 1. kanálu A/D převodníku
37. **PA0**/**ADC0** – Obecný pin portu A nebo vstup 0. kanálu A/D převodníku
38. **VCC** – Vstup napájení +5 V
39. **GND** – Zem pro napájecí napětí
40. **PB0/T0/XCK** – Obecný pin portu B nebo vstup hodin čítače/časovače 0 nebo USART

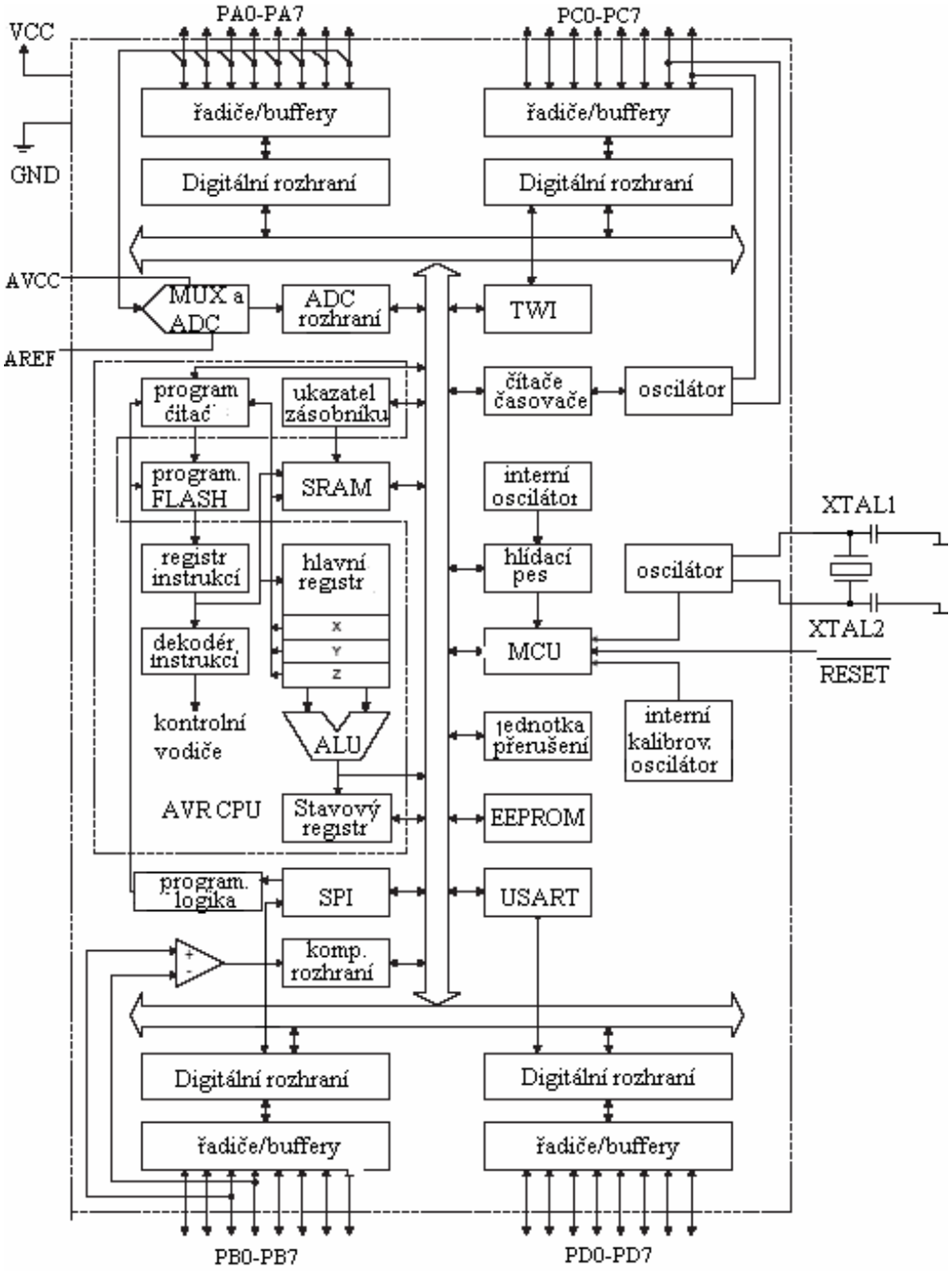
41. **PB1/T1** – Obecný pin portu B nebo vstup hodin čítače/časovače 1

42. **PB2**/**AIN0/INT2** – Obecný pin portu B nebo + vstup komparátoru nebo vstup přeruš. 2

43. **PB1/AIN1/OC0** – Obecný pin portu B nebo – vstup komparátoru nebo výstup čít./čas. 0

44. **PB4**/$\overline{SS}$ – Obecný pin portu B nebo pin, označující Slave zařízení pro rozhraní SPI

2.3.3 Blokové schéma



Obr. 25 Blokové schéma ATmega16

HLAVNÍ ČÁSTI MIKROKONTROLÉRU

Jádro procesoru AVR

Na obr. 25 vidíme, že stěžejní částí procesoru je právě jeho AVR jádro. Jeho prioritní funkcí je zabezpečení bezchybného vykonávání programu. Jádro ovládá všechna přerušení, provádí numerické propočty, má přístup k pamětím a kontroluje periferie. Při vykonávání jedné instrukce je druhá předpřenášena (pre-fetched) z programové FLASH paměti, což dovoluje provádět jeden instrukční cyklus v jednom taktu krystalu.

Modul obsluhující přerušení má vlastní kontrolní registry a používá globální povolovací bit ve stavovém registru, jenž je nadřazen veškerým přerušením. Přerušení mají vektorové uspořádání a nejvyšší prioritu má přerušení, jenž je na nejnižší vektorové adrese.

Při volání podprogramu a během přerušení je návratová adresa programového čítače uložena v zásobníku, který je umístěn v paměti SRAM, a jehož omezení je pouze velikost SRAM.

ALU

Její vysoký výkon spočívá v přímém spojení s 32 všeobecně použitelnými registry. Operace mezi registry lze provádět v každém hodinovém cyklu. Funkce ALU jsou aritmetické, logické a bitové.

Stavový registr

Obsahuje informace o výsledcích vykonaných instrukcí. Tento registr není automaticky ukládán při přerušení ani při návratu z obsluhy přerušení. Ukládání lze však nastavit programově. Stavový registr je definován jako osmibitový. Každý bit má speciální funkci.

Soubor 32 všeobecně použitelných registrů

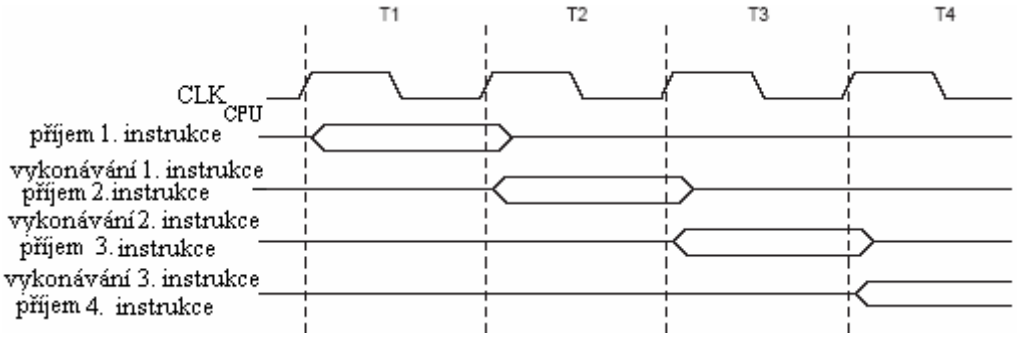
Tento soubor je optimalizovaný pro pokročilé RISC instrukce a je rozdělen do 32 registrů R00-R31. Každému registru je tedy přiděleno osm bitů z datové paměti. Ačkoli registry nejsou fyzicky implementovány jako části SRAM, tak tato koncepce poskytuje velmi dobrou flexibilitu k jejich přístupu. Registry X, Y a Z jsou tři šestnáctibitové registry, vytvořené z párů osmibitových registrů R26, R27 a R28, R29 a R30, R31. Tyto registrové páry slouží jako šestnáctibitové ukazatelé.

Ukazatel zásobníku (Stack Pointer)

Používá se hlavně pro ukládání dočasných dat, lokálních proměnných a návratových adres přerušení či po volání podprogramu. Je definován jako rostoucí z vyšší paměťové části k nižší. Ukazatel vždy ukazuje na vrchol zásobníku. Ukazuje do datové části SRAM, kde jsou uloženy obslužné adresy všech přerušení, proto se tyto obsazené paměťové buňky musí vzít v úvahu a ukazatel zásobníku je nutné nastavit na konec SRAM.

Časování vykonávání instrukcí

CPU je přímo řízen hodinami clk_{CPU}, které jsou generovány vybraným zdrojem hodinového signálu. Obr. 26 ukazuje způsob práce pří vykonávání instrukcí. Jedná se o systém pipeline, sloužící k získání více než 1 MIPS při 1 MHz.



Obr. 26 Časový harmonogram operací s instrukcemi

Reset a ovládání přerušení

AVR jádro poskytuje více zdrojů přerušení. Je k dispozici vektor několika resetů a několika přerušení. Každé přerušení má přiřazeno své přístupové bity. Pokud chceme povolit určité přerušení, musí být příslušné bity zapsány logickou jedničkou a spolu s nimi i globální bit povolení přerušení ve stavovém registru.

Nejnižší adresy programové části SRAM jsou předem definovány jako vektory resetů a přerušení, přičemž nejvyšší prioritu má nejnižší adresa, a ta je vyhrazena pro reset. Druhou nejvyšší prioritu má externí přerušení. Přerušení lze globálně zakázat pomocí instrukce CLI a povolit pomocí SEI. Doba do vykonání přerušení je minimálně čtyři hodinové cykly. Tyto cykly jsou nezbytné pro vyhledání a spuštění požadovaného přerušení.

Paměti

Mikrokontrolér má paměť pro data a program. Navíc má přidanou paměť E^2PROM pro ukládání statických konfiguračních dat. Všechny paměti jsou lineární a regulární.

FLASH programová paměť

Mikrokontrolér má 16 kB programovatelné paměti FLASH pro umístění kódu programu. Je rozdělena na části pro zaváděcí program a pro aplikační programy. Je vyrobena pro až 10 000 přepisů.

SRAM datová paměť

Tato paměť obsahuje celkem 1024 buněk. Prvních 96 je obsazeno speciálními a vstupně-výstupní registry. Ostatní buňky už jsou dostupné pro individuální umístění uživatelských dat. K přístupu k SRAM je možné využít několik možností adresování.

EEPROM datová paměť

K dispozici je 512 B této paměti. Je organizována jako oddělená datová oblast, ve které může být zvlášť každý bajt čten i zapisován. Výdrž je odhadována na 100 000 přepisů. Přístup CPU k této paměti je pomocí speciálních registrů - adresního, datového a kontrolního.

I/O paměť

Všechny vstupně/výstupní zařízení a periferie jsou reprezentovány v I/O části. Vše je přístupné vstupními/výstupními instrukcemi, a to pomocí instrukce přesunu mezi všeobecně použitelnými registry a vstupně/výstupním paměťovým prostorem. I/O registry umístěné v rozsahu paměti 0x00 až 0x1F jsou přímo bitově přístupné pomocí instrukcí SBI a CBI. Hodnoty jednotlivých bitů lze zjistit pomocí instrukcí SBIS a SBIC.

Systémové hodiny a hodinové obvody

Hodinový signál je rozdělen do různých částí obvodu. Pro snížení spotřeby lze tam, kde nejsou používány, tyto hodiny vypnout pomocí spánkových režimů. Externí krystalový oscilátor se připojuje jako u obvodu FT232BM, ale vývody jsou označeny jako XTAL1 a XTAL2. Frekvence tohoto oscilátoru by měla být 8 MHz - je-li CKOPT nenaprogramován a 16 MHz je-li CKOPT naprogramován. CKOPT je jakási pojistka při přepínání mezi dvěma módy zesilování oscilátoru. Externí RC oscilátor se zapojuje na XTAL1, GND a VCC. Používá se pro málo citlivé aplikace. Kapacita kondenzátoru by měla být menší než 22pF. Signál externích hodin se připojuje na vývod XTAL1. Interní kalibrovaný RC oscilátor poskytuje frekvence 1, 2, 4 a 8 MHz. A/D převodník má k dispozici vlastní hodinový okruh, což dovoluje zastavit práci CPU a tím redukovat šum.

Power management a režimy spánku

Pro zapnutí jednoho z šesti režimů spánku musíme zapsat log. 1 do bitu SE v registru MCUCR. Další bity pak definují, jaký typ spánkového režimu bude vykonán.

IDLE – zastaví CPU, ale stále jdou USART, SPI, A/D,…

ADC noise reduction – zastaví především hodiny CPU a FLASH paměti, kvůli redukci šumu

Power-down – zastaví veškeré hodinové signály i oscilátor, pracují jen asynchronní moduly

Power-save – hlavní rozdíl oproti předchozímu je v rychlejším probouzení čipu

Standby – stejný jako Power down, ale oscilátor běží a probouzení je rychlejší

Extended Standby - stejný jako Power save, ale oscilátor běží a probouzení je rychlejší

Integrovaný A/D převodník

Mikrokontrolér obsahuje desetibitový A/D převodník pracující na principu postupné aproximace. Vzorkovací rychlost je 16 kSPS. Rozsah měřitelných vstupních hodnot je od nuly do napětí na vstupu AREF. Měřený signál je možné přivést také na 7 diferenciálních vstupů. Tímto zapojením ovšem ztratíme jeden vstupní kanál. Obvod obsahuje vlastní obvod S/H. Napětí na VCC by mělo být stejné jako na AVCC. Povolení funkce převodníku se děje pomocí bitu ADEN v registru ADCSRA. Výstup převodníku je prezentován v registrech ADCH a ADCL. K započetí převodu se musí zapsat log 1 do bitu ADSC. Start převodu může být také odvozen od jiných zdrojů nebo může být i automatický. Pokud nepožadujeme desetibitové rozlišení, pak vlastní hodinová frekvence převodníku má být vyšší než 200 kHz. První konverze po zapnutí převodu trvá 25 hodinových cyklů, ostatní 13 cyklů. Pro konverzi máme na výběr ze dvou módů - jednorázové konverze a volnoběžný mód. Používáme-li volnoběžnou konverzi, pak po ukončení převodu se okamžitě rozběhne převod další.

Některá důležitá rozhraní

a) TWI rozhraní

Jedná se o dvouvodičové rozhraní. Po jednom vodiči jdou data (pin SDA), po druhém hodinový signál (pin SCL). Případný externí HW by požadoval připojení pull-up rezistoru na každý vodič. Zařízení připojená na tuto sběrnici, lze nakonfigurovat jako Master či Slave. Master zařízení řídí přenos dat. Adresní i datové pakety posílané přes TWI mají 9b délku. Hodinová frekvence pinu SCL, se kterou TWI pracuje, se vypočítá z požadovaných parametrů.

b) USART rozhraní

Toto označení nese univerzální synchronní a asynchronní přijímač a vysílač (Universal Synchronous and Asynchronous Receiver and Transmitter). Skládá se ze tří základních částí: generátor hodin, přijímač a vysílač. Vysílač obsahuje buffer, sériový posuvný registr a kontrolní logiku pro různé datové formáty. Hodinový signál slouží pro synchronizaci jednotlivých částí a operací.

Přijímač je složen z obvodu pro kontrolu parity, kontrolní logiky a dvouúrovňového přijímacího bufferu. Tímto bufferem se liší od starší verze - UART. Ve verzi USART může posuvný registr navíc pracovat jako další buffer.

USART se shoduje s UART v těchto bodech:

- ✓ umístění bitů v registrech
- ✓ generátor přenosové rychlosti
- ✓ vysílací operace
- ✓ funkčnost vysílacího bufferu
- ✓ přijímací operace

Data jsou přenášena tak, že nejdříve je vyslán START bit, potom 5 až 9 datových bitů (počínaje LSB a MSB konče), pak kontrolní bit parity (nemusí být) a 1 či 2 STOP bity. Přerušení jsou volána při vyprázdnění datového registru a při ukončení příjmu či vysílaní.

USART může také pracovat v asynchronním režimu. Operace v tomto režimu zajišťuje logika, která asynchronně přijímaná data ovzorkuje, synchronizuje a zpracuje. Každý přijímaný bit je ovzorkován a filtrován přes dolní propust kvůli rušení. Operační rozsah přijímaných dat závisí na přesnosti generátoru přenosové rychlosti, na rychlosti příchozích dat, apod.

Před startem USART je nutná inicializace. Nejdříve je třeba zakázat všechna přerušení. Inicializace dále obnáší nastavení přenosové rychlosti, formátu dat a nastavení USART jako přijímače (bit TXEN) či vysílače (bit RXEN). TXC Flag slouží pro zjištění ukončení vysílání a RXD Flag pro ujištění, že byla přečtena veškerá data vstupního bufferu.

c) SPI rozhraní

Jde o sériové rozhraní pro periferie. Dovoluje vysokorychlostní komunikaci mezi mikrokontrolérem a ostatními zařízeními. Je také možné spojení dvou mikrokontrolérů, jednoho jako Slave a druhého jako Master. Systém přenosu dat spočívá v použití posuvných registrů (v obou spojených zařízeních) a generátoru hodinové frekvence od zařízení nastaveného jako Master. Je na výběr ze sedmi různých přenosových rychlostí. Vysílací buffer je jednoduchý, přijímací je dvouúrovňový.

Pokud je SPI nastaveno jako Slave, pak pin $\overline{SS}$ (Slave Select) slouží jen jako vstup. Je-li tento pin na log. 0, pak je SPI aktivováno a pin MISO (Master In Slave Out) se stane výstupním (pokud je toto nastaveno uživatelem). Všechny ostatní piny jsou potom vstupy. Je-li $\overline{SS}$ na vysoké úrovni, pak všechny piny jsou vstupní a SPI je nečinné, což znamená, že SPI nebude přijímat přicházející data. Je-li SPI nastaveno jako Master (bit MSTR v SPCR registru), pak si uživatel může vybrat, jestli pin $\overline{SS}$ bude vstupní či výstupní. Bude-li tento pin nastaven jako vstupní, pak nebude mít vliv na SPI systém a bude řízen pinem $\overline{SS}$ zařízení, uvažovaného jako Slave. Bude-li nastaven jako vstupní, pak musí být držen na vysoké úrovni, aby mohl provádět Master SPI operace.

3. Popis HW části zařízení

3.1 Vlastní popis zapojení

Celé zařízení je postaveno na jedné desce plošných spojů. Napájení všech obvodů je realizováno ze sběrnice USB, čímž se ušetří externí napájecí zdroj a další přívodní kabel. Vstup měřeného napětí je proveden pomocí standardního konektoru BNC. Na přední straně zařízení jsou kromě BNC konektoru také tři LED diody, indikující přítomnost napájecího napětí a příjem nebo vysílání dat z PC. Na zadní straně je konektor USB a dvě drobná tlačítka. První slouží pro resetování procesoru, druhé resetuje A/D převodník. Při normálním používání přístroje by neměly nikdy být použity. Jsou určeny pro krajní řešení situací, při kterých by se zařízení začalo chovat nestandardně nebo pokud byl počítač přepnut do úsporného režimu a zařízení nebylo odpojeno.

Důvodem použití sběrnice USB jako komunikačního portu ze zařízením je především jeho hojné rozšíření, vysoká přenosová rychlost a existence obousměrných převodníků ze sériové linky mikrokontroléru na USB, tedy UART/USB. Jinými počítačovými porty, jako např. sériovými nebo paralelními se např. v současné době už nové notebooky ani neosazují. Port fireware zase není tak rozšířený. Zařízení pracuje s USB 2.0. Starší počítače (nikoli extrémně zastaralé) obsahují alespoň pomalejší verzi USB. Tato verze, označovaná USB 1.1, je plně kompatibilní s novější verzí USB 2.0, pouze s rozdílem, že maximální přenosová rychlost je u první verze 12 Mb/s, zatímco u novější je to 480 Mb/s. Z uvedeného tedy plyne, že zařízení pracuje i se starší verzí USB, protože není využito ani její plné rychlosti.

Protože popisované zařízení je pouze prvním přiblížením k danému problému, a protože A/D převodník má dostatečně vysoký vstupní odpor, není před jeho vstup zařazen předzesilovač. Navíc takový vstupní předzesilovač by v našem případě bylo nutné napájet souměrným napětím alespoň +5 V a –5 V. Napájecí napětí celého zařízení je +5 V, jenž je získáváno z USB. Záporné napájecí napětí pro předzesilovač by bylo nutné získat z kladného, nejlépe pomocí DC/DC měniče (např. MC34063). Pak by však bylo nutné přidat na plošný spoj nejen tento měnič, ale i pomocné satelitní součástky, stabilizátor výstupního záporného napětí a filtrační kondenzátory s dostatečně vysokou celkovou kapacitou. Celkové zapojení by se poměrně rozšířilo a navíc by pravděpodobně nebylo možné napájet zařízení z USB z důvodu vysoké proudové spotřeby. Popisované schéma je na obr. 27.

Vstupy A/D převodníku jsou tedy pouze chráněny oboustranným napěťovým omezovačem. Vstupní omezovač je vytvořen z antisériového spojení dvou zenerových diod, zapojených mezi vstupy A/D převodníku. Maximální vstupní rozsah měřitelného napětí je +2,5 V.až –2,5 V. Proto jsou použity ochranné diody D5 a D6 se zenerovým napětím 2,7 V, což je v podstatě nejnižší standardně vyráběná hodnota zenerova napětí. Lépe by samozřejmě bylo lepší použít diody s ještě nižším zenerovým napětím, protože diodový omezovač začíná pracovat až při napětí mezi vstupy vyšším, než je 3,4 V (2,7 V + 0,7 V). Vstup A/D převodníku je však také chráněn nízkovýkonovými diodami, jenž jsou v něm integrovány. Tyto vestavěné diody budou ořezávat vyšší napětí na vstupu, než je 2,5 V. Při příchodu vyšší napěťové špičky by už byla amplituda vstupního napětí oříznuta zenerovými diodami.

Na každém vstupu je ještě připojen RC článek (R2 C1 a R1 C2), který slouží jako horní zádrž. Hodnoty součástek jsou voleny tak, aby mezní kmitočet tohoto článku byl nad maximálním měřitelným kmitočtem. Takto se nám do A/D převodníku zbytečně nedostávají rušivé signály, které bychom s popisovaným přístrojem stejně nemohli změřit. A/D převodník vzorkuje maximální rychlostí 100 kSPS, proto je RC článek navrhnut tak, aby pokles 3 dB nastal přibližně na kmitočtu 100 kHz. Aby byla splněna vzorkovací věta, je možné spolehlivě měřit signály jen asi do 20 kHz (50 kHz).

K A/D převodníku je nutné připojit referenční napětí o amplitudě 2,5 V. To je získáno pomocí obvodu TL431. Jeho připojení je velice snadné a navíc vyžaduje pouze dvě externí součástku – blokovací kondenzátor výstupního napětí 2,495 V a omezovací rezistor, zapojený mezi napájecí napětí a jeho vstup. Pro vyhlazení vstupního proudu slouží tlumivka L1.

Napájecí napětí A/D převodníku je filtrováno a blokováno dvojicí kondenzátorů C5 a C20. Tyto součástky je nutné umístit co nejblíže k převodníku. A/D převodník musí mít napájenou i digitální část. Její napájení je odebíráno s analogového napájení, je však filtrováno článkem R4 a C6. Navíc je připojen i tantalový kondenzátor pro úplné vyhlazení napětí. Resetování A/D převodníku lze provést pomocí RC článku, kde kondenzátor C3 je vybíjen paralelně zapojeným tlačítkem. Resetovacím signálem je úroveň logické jedničky po určitou dobu, která je bohatě pokryta stiskem tlačítka. Převod A/D převodníku je spouštěn signálem logické nuly na vstup CNVST, jenž je proto připojen k mikrokontroléru. Výstupní data A/D převodníku jsou k dispozici na celkem 16 pinech, jenž jsou připojeny na dva osmipinové porty mikrokontroléru.

Mikrokontrolér je také napájen napětím +5 V a musí mít také připojené blokovací kondenzátory. Přes tlumivku L2 je napájen také jeho vnitřní ADC, ten ale používán nebude.

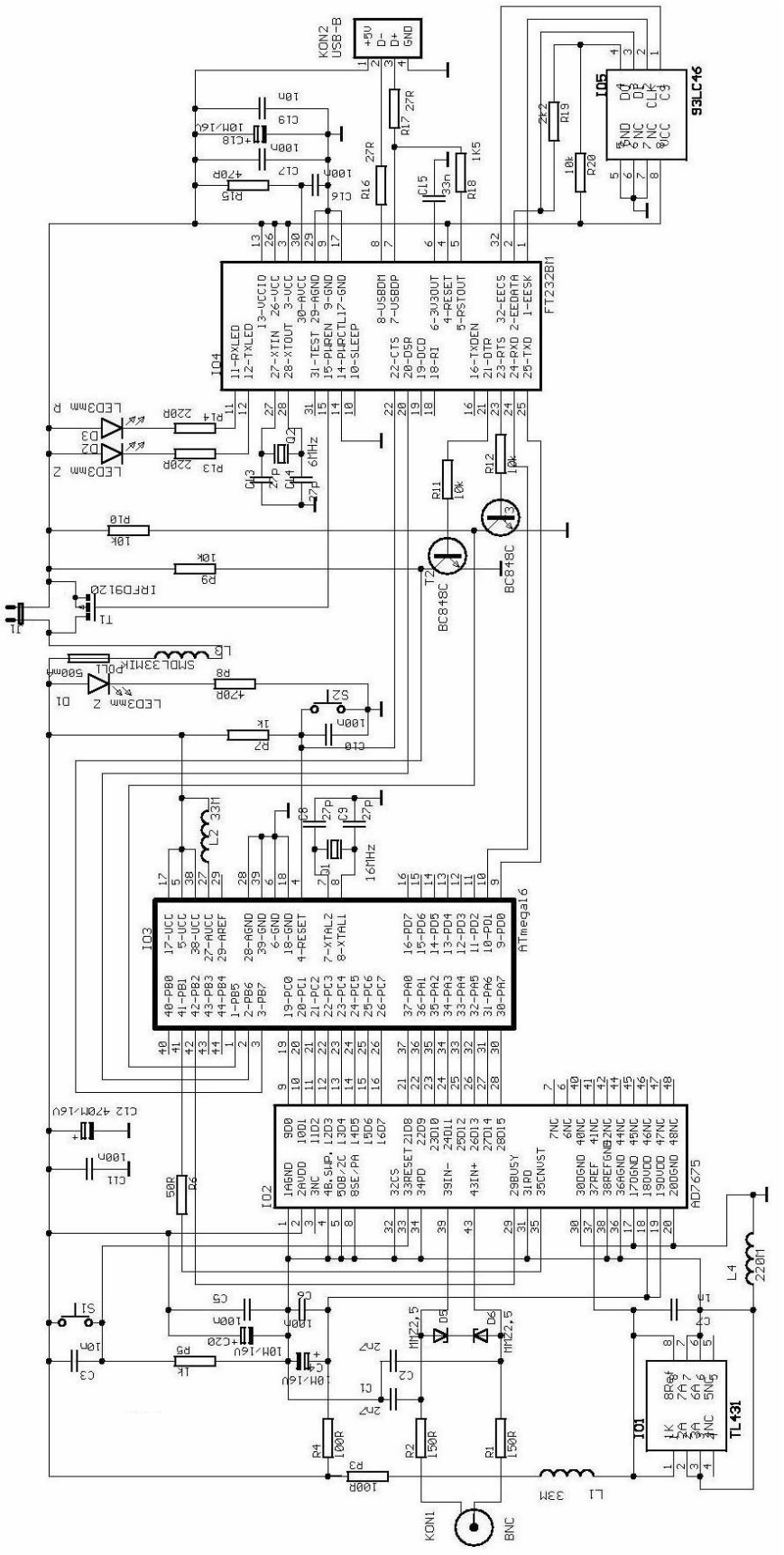
Jako zdroj hodinového kmitočtu pro mikrokontrolér je použit krystal 16 MHz, který je blokován keramickými kondenzátory. Resetování mikrokontroléru je řešeno pomocí obvodu FT232BM. Navíc lze reset mikrokontroléru provést i pomocí RC článku s tlačítkem.

Komunikační sériová jednotka mikrokontroléru SPI je přístupná na pinech PB5, PB6 a PB7. Tato jednotka je připojena k obvodu FT232BM. Vývod PB6 je spojen s tímto obvodem přímo, ostatní dva piny jsou řízeny prostřednictvím tranzistorů. Toto spojení je využíváno pro sériové programování mikrokontroléru. Komunikace s programem v počítači je realizována pomocí asynchronně pracující jednotky UART, jenž je také v mikrokontroléru integrovaná. Komunikace probíhá pouze po dvou linkách, jedna data přijímá – RXD, druhá vysílá – TXD. Tyto vývody mikrokontroléru jsou spojeny s příslušnými piny převodníku FT232BM. Aby nebyl nikdy překročen maximální proudový odběr z USB (500 mA), je do obvodu zařazena vratná pojistky – polyswitch, dimenzovaný na 500 mA. Napájecí napětí je vedeno přes tlumivku L3, jenž do jisté míry vyhladí průběh proudu a tím zabrání rušení A/D převodníku šumem z počítače.

Všechny obvody, kromě FT232BM a paměti EEPROM, jsou připojené přes spínací tranzistor MOSFET. Jelikož je však proudový odběr zařízení cca na hranici 100 mA, vzniká nebezpečí, že tranzistor nebude sepnut trvale, právě kvůli spotřebě vyšší než 100 mA. Proto je k tranzistoru připojen jumper, který slouží pro případ, že programovací aplikace nenalezne mikrokontrolér. Toto je způsobeno neotevřením tranzistoru a tedy nepřivedením napájecího napětí na mikrokontrolér. V takovém případě je nutné jumper zapojit. Obvod FT232BM má pro spínání tranzistoru speciální vývod. Sepnutí tranzistoru je indikováno rozsvícením LED diody, jenž je vyvedena na přední panel zařízení. K obvodu FT232BM je nutné připojit krystal 6 MHz, spolu s blokovacími kondenzátory. K dispozici jsou také speciální výstupy pro LED diody, které pak signalizují režim přijmu nebo vysílání dat do počítače. Tento obvod je napájen, na rozdíl od ostatních, přímo z USB a jeho napětí je opět blokováno. Výstup 3V3OUT není použit a proto je pomocí kondenzátoru blokován proti zemi. Integrovaná násobička hodinové frekvence je napájena přes rezistor a také blokována (vstup AVCC). Datové výstupy jsou na USB vedeny přes ochranné rezistory R16 a R17. Výstup RSTOUT je dle doporučeného zapojení spojen přes rezistor k vývodu USBDP. K příslušným vývodů je připojena externí paměť.

Externí paměť typu EEPROM je připojena podle doporučeného zapojení. Její funkce bude spočívat v tom, že bude obsahovat důležité informace pro USB. Tedy například jednoznačné identifikační číslo, informaci o proudové spotřebě, název, apod.

3.2 Schéma celého zařízení



Obr. 27 Schéma celého zařízení

4. Popis ovládacích programů

4.1 Program pro mikrokontrolér

Program pro mikrokontrolér byl vytvořen v AVR Studiu® 4, jenž je volně distribuován. AVR Studio® 4 je kompletní vývojové prostředí, ve kterém lze program napsat, zkompilovat i ladit. Prostředí je optimalizováno pro Windows® 9x/NT/2000 a XP. Úplný výpis kódu programu, který bude dále popsán, se nachází v příloze č. 6.

Direktivy – začátek kódu programu

Úplným začátkem programu jsou direktivy. Direktiva .NOLIST sděluje překladači, aby následující kód nevypisoval do výstupního souboru *.LST. Důvod jejího použití bude vysvětlen záhy. Další direktivou je .INCLUDE, která do zdrojového textu implementuje soubor m16def.inc. Tento soubor obsahuje přiřazení názvů ke všem registrům daného procesoru. Pak už nemusíme registry adresovat jejich hexadecimální adresou, nýbrž můžeme použít zavedeného jména, což je podstatně přehlednější. Tento soubor obsahuje asi 800 takových definičních přiřazení. Jeho výpis nás vlastně nezajímá, a proto byla použita direktiva .NOLIST. Další zdrojový kód už je pro nás důležitý a výpis do *.LST je proto zase aktivován pomocí direktivy .LIST. Řádkem .EQU BAUD=8 přiřadíme symbolu BAUD hodnotu osm. Tato hodnota bude po nahrání do příslušného řídícího registru určovat přenosovou rychlost jednotky UART, číslo osm odpovídá rychlosti 115,2 kBd. Direktiva .DSEG říká překladači, že další direktivy se budou týkat datové paměti RAM. Další řádek – .ORG $60 znamená přesun programového čítače na adresu 60h v datové paměti. Od této adresy si připravíme prostor pro uložení naměřených dat. Nelze začínat od adresy 0h, protože zde je umístěno 64 vstupně/výstupních a 32 univerzálních registrů ($64+32=96_{10}$ ~ 60h). Další direktivou je .BYTE (802). Její funkce spočívá ve vyhrazení prostoru v datové paměti o délce 802 bajtů. Jeden naměřený a uložený vzorek představuje dva bajty a celkem je odebíráno 401 vzorků. První bajt je tedy vyhrazen na adrese 0060h (96_{10}) a poslední na adrese 0381h ($96_{10}+801_{10}=897_{10}$). Před touto direktivou je ještě návěští DATA. To nám umožní adresovat datový prostor 802 bajtů jeho jménem – DATA. Nebude tedy nutné nepohodlně přepočítávat adresu požadované buňky, abychom zjistili její pořadové číslo. Direktiva .CSEG sdělí překladači, že další instrukce se budou týkat programové části paměti.

Návěští RESET

První operací, kterou je bezpodmínečně nutné provést, je vymezení správné oblasti, ve které se může pohybovat ukazatel zásobníku (Stack Pointer, SP). Touto oblastí je datová paměť RAM. Interně je totiž SP nastaven na adresu 0000h. Kdybychom si v této konfiguraci uložili třeba návratovou adresu do RAM, pak by se SP dostal na adresu FFFFh, která je však mimo rozsah interní RAM, jež končí na 045Fh. SP se totiž při uložení dat snižuje. Z tohoto důvodu musíme SP nastavit na konec datové paměti RAM. Pokud bude tedy SP na konci RAM, pak při uložení informace je jeho hodnota snížena na předposlední adresu v RAM. SP je šestnáctibitový registr, a je tedy rozdělen na dva osmibitové. Dolní bajt ukazatele zásobníku představuje registr SPL (Stach Pointer Low), horní pak SPH (Stach Pointer High).

Nastavení ukazatele zásobníku je provedeno pomocí registru R16. Nejprve je tedy do R16 načten dolní bajt adresy konce RAM. Je použit symbol RAMEND, který představuje poslední platnou adresu RAM – 045Fh. Dolní bajt této adresy (5Fh) je získán funkcí LOW. Takto získaná dolní část adresy konce datové paměti je z registru R16 poslán do dolního bajtu ukazatele zásobníku - SPL. Pak je získán horní bajt ze symbolu RAMEND – 04h a nahrán do registru R16. Následně je tato hodnota nahrána do SPH.

Jelikož pro přenos naměřených dat používáme sériovou jednotku UART, je nutné nastavit požadovanou rychlost přenosu dat a také formát přenášených dat. Nastavení přenosové rychlosti se provede nahráním konstanty do registru UBRR. Tuto hodnotu je nutné vypočítat podle vzorce (9), na základě požadované přenosové rychlosti a frekvenci krystalu mikrokontroléru. Po dosazení nám vyjde hodnota 7,68. Jelikož desetinné číslo nelze zapsat do registru, musíme hodnotu zaokrouhlit na nejbližší celé číslo, tedy na 8. Proto jsme na začátku přiřadili symbolu BAUD právě hodnotu osm. Registr UBRR je šestnáctibitový a je tedy rozdělen na dva osmibitové. Horní bajt je UBRRH a dolní tedy UBRRL.

$$UBRR = \frac{f_{krystalu}}{16 * f_{požadovaná}} - 1 = \frac{16 * 10^6}{16 * 115{,}2 * 10^3} - 1 = 7{,}68 \approx 8 \qquad (9)$$

Vlastní nastavení požadované přenosové rychlosti provedeme jednoduše. Do registru R16 si nahrajeme dolní bajt symbolu BAUD, tedy vlastně dolní bajt z 0008h. Obsah R16 pak nahrajeme do UBRRL. Pak si do R16 nahrajeme horní bajt z BAUD a následně obsah registru přesuneme do UBRRH. V podstatě by stačilo do UBRRL nahrát 08h a do UBRRH 00h. Výhodou zdlouhavého zápisu hodnot do UBRR je snadná případná změna přenosové rychlosti UART. Stačí pouze za direktivou .EQU BAUD=8 dosadit místo 8 jiné číslo.

Také je nutné aktivovat funkce přijímání a vysílání dat jednotkou UART. To jednoduše provedeme instrukcí LDI R16,(1<<RXEN)|(1<<TXEN). Tento řádek nastaví třetí a čtvrtý bit registru R16 na jedničky, ostatní bity vynuluje. Překladač totiž v souboru m16def.inc zjistí, že symboly RXEN a TXEN odpovídají trojce a čtyřce. My se už o to zajímat nemusíme. Takto nastavený registr R16 nahrajeme do řídícího registru UCSRB sériové jednotky. Posledním nutným nastavením UART je počet přenášených bitů a typ režimu. Zvolíme asynchronní režim a osmibitový přenos s jedním stop bitem a bez parity. Opět nastavíme příslušné bity v R16 a následně tuto hodnotu přeneseme do dalšího řídícího registru, do UCSRC.

Nastavení vstupních a výstupních portů

Před použitím portu je nutné nastavit port jako vstupní nebo výstupní. Všech osm pinů jednoho portu však nemusí být ve stejném režimu. Na jednom portu tedy můžeme mít například čtyři piny vstupní a čtyři výstupní. O režimu jednotlivých pinů rozhoduje registr DDRx, kde x může být A, B, C nebo D.

V našem případě bude používán kompletní port A i port C jako vstupní, proto do registrů DDRA a DDRC nahrajeme hodnotu odpovídající vstupnímu režimu, tedy nulu. Instrukcí CLR vynulujeme R16 a jeho hodnotu pošleme do požadovaných registrů. Port B bude výstupní, pouze bit 2 bude sloužit jako vstupní. Proto do registru pro řízení režimu portu B nahrajeme nejdříve hodnotu FFh a následně v tomto registru vynulujeme bit 2. Instrukcí SER nastavíme všechny bity registru R16 na jedničky a pomocí instrukce OUT toto číslo zkopírujeme do registru DDRB. Řádek kódu – CBI DDRB, 2 nám vynuluje druhý bit zmiňovaného portu. Pro správnou funkci sériové linky je nutné jako výstupní nastavit pin 1 portu D. Toto nejrychleji provedeme zápisem SBI DDRD, 1.

Další pomocná nastavení

Protože větší počet instrukcí pracuje s hodnotami uložených v registrech, než s neuloženými konstantami. Je lepší si hned na začátku do některých registrů uložit často používané konstanty. Jinak bychom je v případě potřeby museli nahrát do nějakého registru a teprve po té s nimi provést nějakou operaci. Tím bychom zbytečně zpomalovali chod programu. Naproti tomu nevýhodou trvalého uložení nějaké konstanty je ztráta jednoho registru, jakožto ukládacího prostoru.

Často používaná bude konstanta FFh (255_{10}), proto si ji nahrajeme instrukcí LDI do registru R21. Další důležitá hodnota je pro nás F0h (128_{10}), ta je umístěna do registru R20. Poslední konstantou je 00h, jež bude trvale v registru R19.

Poslední operací, která bude provedena pouze po startu mikrokontroléru, je zakázání startu převodu A/D převodníku. To provedeme tímto řádkem: SBI PORTB, 1. Na výstupním portu B se bit 1 dostane do stavu logické jedničky. Tento pin je přiveden na vývod $\overline{CNVST}$ použitého A/D převodníku. Přivedená logická jednička způsobí zakázání startu převodu, až do příchodu logické nuly. Tato operace slouží jako ujištění, že při pozdějším, úmyslném startu převodu nebude mikrokontrolér muset čekat, než se dokončí předchozí převod, jehož výsledek jsme ani nemohli zaznamenat.

Hlavní smyčka programu

Tato část programu bude vykonávána více méně ve smyčce. Až do této chvíle nás nemusela zajímat časová náročnost každé instrukce, protože se doposud jednalo o část inicializační, která se při standardním chodu programu provede pouze jedenkrát. Výjimkou je pouze první část hlavní smyčky – čekání na povel z počítače.

Smyčka CEKEJ_PRIKAZ

Start programu je odvozen od příchodu jakéhokoli bajtu z počítače. Počítač si tímto způsobem říká o data. Na tomto místě bychom mohli použít přerušení. Ukazuje se však, že v tomto konkrétním místě by to nebylo výhodné. Přijetí znaku je indikováno logickou jedničkou příznakového bitu RXC v registru UCSRA. Jelikož je tento bit na sedmém, nejvyšším místě v registru UCSRA, tak hodnota tohoto registru bude minimálně F0h. Tato hodnota je minimální proto, že může být nastaven i některý další bit v tomto registru. My si proto hodnotu tohoto registru načteme pomocí instrukce IN do registru R16. Další instrukcí ANDI odmaskujeme v R16 spodních sedm bitů, tedy vynulujeme všechny bity až na hlídaný sedmý. Po této operaci bude mít R16 jen dvě možné hodnoty: 00h nebo F0h. Druhou hodnotu nabude v případě, že z počítače byl vyslán požadavek pro přenos dat. Instrukcí CPSE porovnáme náš R16 s další registrem R20, ve kterém jsme si na začátku uložili právě hodnotu F0h. Pokud budou hodnoty v R16 a R20 různé, pak se bude normálně pokračovat na instrukci RJMP, která vrátí tok programu na začátek smyčky CEKEJ_PRIKAZ. V tomto případě totiž nepřišel žádný požadavek z počítače, a tak musí mikrokontrolér čekat, dokud takový nepřijde. V opačném případě, tedy pokud hodnota v R16 bude shodná s hodnotou v R20, pak došlo k příchodu požadavku a program tedy může pokračovat dále, díky instrukci CPSE.

Návěští AD – příprava na ukládání vzorků

Nejprve je instrukcí IN do registru R25 přečtena hodnota přijímacího bufferu UDR sériové linky UART. Tato hodnota byla přijata z počítače a určuje počet průchodů zpomalovací smyčkou, která slouží pro časování odběru vzorků, tedy řídí vzorkovací rychlost.

Abychom mohli jednoduše ukládat naměřená data do vyhrazené oblasti DATA v datové paměti RAM, použijeme ukazatel. Registry R31 a R32 slouží jako šestnáctibitový ukazatel, jenž byl nazván Z a je složen z dolního bajtu ZL (R31) a z horního ZH (R32). Do ZL si tedy pomocí instrukce LDI a funkce LOW nahrajeme dolní bajt první adresy, od které budeme ukládat naměřené vzorky. Horní bajt této adresy načteme do registru ZH.

Použitý A/D převodník je šestnáctibitový, jeden vzorek tedy představuje 16 bitů čili dva bajty. Pro uložení jednoho vzorku tedy bude potřeba dvou paměťových buněk. Pro uložení 401 vzorků budeme tedy potřebovat vyhradit 802 bajtů v paměti RAM. To jsme provedli již na začátku kódu programu. Zde zcela oprávněně vyvstává otázka: Proč zrovna 401 vzorků? V podstatě máme v datové paměti k dispozici 1024 bajtů volného prostoru, z čehož by vyplýval počet možných uložených vzorků 512. Je však nezbytně nutné nepoužívat koncovou část RAM, protože v této oblasti bude ukazatel zásobníku ukládat návratové adresy. Kdybychom zde tedy měli uloženy naměřené hodnoty, tak by v lepším případě byly pouze ukazatelem zásobníku přepsány. V horším případě by mohla být naše data brána jako návratová adresa a tok programu by byl nesmyslně nasměrován na náhodné místo v programu. Proto bylo zvoleno 400 vzorků plus jeden vzorek navíc, kvůli technice zobrazení na obrazovce počítače.

Pro přesné odměření 401 vzorků je třeba si zřídit nějaké počítadlo. Nejjednodušší je naplnit registr na požadovaný počet vzorků a po každém průchodu jeho hodnotu dekrementovat. Ve smyčce by pak byl umístěn test nulovosti tohoto registru a v případě nuly by se smyčka opustila. Na stejném principu je založeno počítadlo v programu. Jelikož ale hodnotu 401 nelze uložit do jednoho registru, jsou použity registry dva (R17, R18). Do prvního je nejprve uloženo číslo 252, do druhého pak hodnota 150. Ano, součet nedává požadovanou hodnotu 401, ale 402. To je nutné, kvůli způsobu práce programu. My tedy nejdříve naplníme dva počítací registry, ty jsou postupně odečítány. Až budou mít oba registry nulové hodnoty, provedou se další operace. Program je poté vrácen zpět, jsou znovu naplněny počítací registry a pokračuje se opět dále ve smyčce načítání dat. Když jsou registry počítadla poprvé naplněny, můžeme konečně spustit převod A/D převodníku. Od tohoto okamžiku bude A/D převodník neustále pracovat.

Návěští CTI – příprava časování vzorkování

Nejprve si zkopírujeme již uloženou hodnotu v R25 do registru R29. To proto, hodnota R25 slouží k časování odběru vzorků. Kdybychom pomocný R29 nepoužívali, pak bychom museli dekrementovat R25. Po odběru jednoho vzorku by už byl R25 nulový a nebylo by možné správně načasovat odběr dalšího vzorku. Proto si jeho hodnotu uložíme do pomocného R29 a tento dekrementujeme. Po odběru každého vzorku jeho původní hodnotu zase obnovíme, zkopírováním z R25.

Snímání naměřených dat

A/D převodník má jeden kontrolní výstup – pin BUSY, kterým je indikováno dokončení převodu a tedy i připravenost převedených dat ke čtení. Před čtením dat z jeho výstupů bychom tedy měli počkat, až tento pin bude ve stavu logické nuly. To však v našem případě není nutné, protože časování odběru vzorků je nastaveno minimálně na 10 μs, což odpovídá plné rychlosti vzorkování 100 kSPS. Testováním bylo dokázáno, že takto lze situaci s úspěchem řešit.

Jelikož jsme potřebná opatření pro splnění podmínek pro korektní odběr naměřených dat splnili, sejmutí převedené binární hodnoty z výstupů A/D převodníku bude velice jednoduché. Na port A jsou napojeny výstupy A/D převodníku, na kterých je po dokončení převodu dostupný horní bajt výsledku. Port C je připojen na dolní bajt. Nejprve tedy instrukcí IN sejmeme stav portu A, tím získáme horní bajt převáděné hodnoty. Tato hodnota je uložena do registru R23. Stejným způsobem je sejmut i dolní bajt, jeho stav je uložen do registru R22.

Korekce některých vzorků kvůli synchronizaci

Analogově-digitální převodník poskytuje dva režimy výstupních dat. Prvním režim je normální binární výstup. Druhým režimem je výstup ve dvojkovém doplňku. Pro naše účely byla vybrána druhá možnost. Maximální kladná hodnota jednoho bajtu v desítkové soustavě je ve dvojkovém doplňku rovna 127 ($0111\ 1111_2$). Maximální záporná je pak -128 (1000 0000). Vidíme, že rozsah je pro záporné hodnoty o jednu hodnotu vyšší. Proto je maximální záporná hodnota -128 (ve dvojkovém doplňku) použita pro synchronizační účely, jak uvidíme později. V normálním binárním vyjádření má toto číslo hodnotu 128. A tuto hodnotu jsme si na začátku programu uložili do registru R20. Záměrem je tedy eliminace hodnoty 128 respektive -128 z naměřených hodnot. Pokud tedy horní nebo dolní bajt výsledu nabude této hodnoty, bude zachycen, dekrementován a poslán dál hlavním vláknem programu.

Nejprve je instrukcí CPSE porovnána hodnota dolního bajtu s číslem 128 v pomocném registru. Pokud bude dolní bajt různý od 128, není třeba cokoli provádět a program pokračuje dál pomocí instrukce RJMP. Pokud má dolní bajt hodnotu právě 128, instrukce CPSE způsobí přeskočení následující instrukce – RJMP a prováděn je řádek INC R22. Výše bylo napsáno, že při zachycení čísla 128 v některém bajtu výsledku převodu se musí tato hodnota snížit o jedničku. Ve výpisu programu je však použita instrukce inkrementace, tedy naopak zvýšení čísla 128 o jedničku. Proč? Kód programu je v souladu s požadovanou funkcí. Zdánlivý rozpor je způsoben načítáním bajtů výsledku ve dvojkovém doplňku. Pokud v normální binární soustavě chceme snížit hodnotu čísla o jedničku, pak použijeme instrukci pro dekrementaci. Chceme-li z čísla -128 vytvořit číslo -127, nemůžeme použít instrukci pro dekrementování. Číslo -128 je dvojkovým doplňkem k číslu +128, avšak číslo o jedničku menší, tedy -127, už není doplňkem k +127, ale k hodnotě +129, tedy naopak k číslu o jedničku větším. Vysvětlení snad podává tab. 1.

Standardní zobrazení			**Dvojkový doplněk**	
Binárně	**Dekadicky**		**Binárně**	**Dekadicky**
1111 1111	255		0111 1111	127
1111 1110	254		0111 1110	126
…	…		…	…
1000 0001	129		0000 0001	1
1000 0000	128		0000 0000	0
0000 0001	127		1111 1111	-1
…	…		…	…
0000 0001	1		1000 0001	-127
0000 0000	0		1000 0000	-128

Tab. 1 Porovnání standardního zobrazení s dvojkovým doplňkem

Stejnou operaci musíme provést i s horním bajtem naměřeného výsledku. Pouze v případě, že hodnota horního bajtu výsledku je různá od 128, je proveden skok na jiné návěští, aby program neběžel v nekonečné smyčce.

Po průchodu programu touto částí máme jistotu, že ani jeden z načtených bajtů nemá hodnotu 128. Můžeme tedy přistoupit k dalším operacím s naměřenými daty.

Uložení dat do paměti RAM mikrokontroléru

Uložení naměřených dat se provede instrukcí ST. Dolní bajt výsledku převodu provedeme zápisem řádku: ST Z+, R22. Toto je speciálně použitá instrukce, která uloží horní bajt z registru R22 do paměťové buňky vnitřní paměti RAM, jejíž adresa je určena aktuální hodnotou ukazatele Z. Navíc je tento ukazatel použit v kombinaci s inkrementačním operátorem.

Po každém uložení nějakého bajtu do paměti RAM je ukazatel tímto operátorem inkrementován. Po uložení horního bajtu je uložen dolní bajt, jenž byl prozatím načten v registru R23.

Když máme v RAM uloženy oba bajty výsledku převodu, mohli bychom program vrátit zase k místu, kde byla data načítána z A/D převodníku. Jelikož ale potřebujeme nějakým způsobem řídit dobu mezi odebráním sousedních vzorků, je nutné v tomto místě zavolat rutinu, která toto zajistí. Rutina je umístěna od návěští CEKEJ. Po uložení dat je tato rutina zavolána instrukcí RCALL. Než tok programu přesuneme do části načítání dalšího vzorku, musíme snížit počítadlo vzorků (registry R17 a R18), kterými odpočítáme 401 vzorků. Nejprve je dekrementován R17, pokud by jeho hodnota byla jedna, pak po dekrementaci dostaneme nulu, což by zachytila následující instrukce podmíněného skoku BREQ. Nastane-li tato situace, pak zmíněná instrukce skoku přesune tok programu na návěští DALSI_DEC, kde by byl dekrementován druhý počítací registr – R18. Na toto návěští by byl program vracen po každém uložení vzorku, dokud i R18 nebude nulový. Bude-li R17 i R18 nulový, bude program přesunut zpět na návěští CTI, kde bude načten další vzorek z A/D převodníku.

Návěští DALSI_DEC – odpočítání uložení vzorků

Do této části se dostaneme, pokud má první počítací registr (R17) nulovou hodnotu, to znamená, že už bylo uloženo nejméně 150 vzorků. Po opuštění této části bude ale opět R17 dekrementován a testován na nulovou hodnotu. Pokud bychom však dekrementovali nulový registr, obdržíme hodnotu 255, což by v důsledku vedlo odečítání nekonečného počtu vzorků a k úplnému zaplnění vnitřní datové RAM a tedy i k přepsání hodnot návratových adres ukazatele zásobníku, což by pravděpodobně způsobilo kolizi programu. Proto je při každém průchodu touto částí přiřazena registru R17 hodnota 1, jejíž dekrementování a testování způsobí, při odběru dalšího vzorku, skok programu do této části programu. Pokud je tento problém ošetřen, pak o jedničku snížíme hodnotu druhého počítacího registru R18. Tento registr je pak pomocí instrukce CPSE testován na nulovou hodnotu.

Pokud je hodnota R18 různá od nuly, pak se provede následující instrukce relativního skoku RJMP a tok programu je vrácen na návěští CTI, tedy bude proveden odběr dalšího vzorku, jelikož ještě nebylo odebráno všech 401 vzorků. Bude-li tedy dokončen odběr všech vzorků, pak bude hodnota v R18 nulová a instrukce CPSE přeskočí následující instrukci a bude proveden skok na návěští POSLIDATA, kde budou naměřená data odeslána sériovou linkou.

Příprava vysílání dat přes UART do počítače

Pro přenos dat do počítače si opět musíme zřídit speciální počítadlo, kterým přesně nastavíme počet vysílaných vzorků na 401, přičemž jeden vzorek tvoří dva bajty. Situace je tedy stejná jako při načítání a ukládání vzorků v předchozí části kódu.

Do počítacích registrů R17 a R18 si načteme hodnoty, jejichž součet je roven 402. Jelikož pro adresaci paměťových buněk používáme ukazatel, je nutné si jej správně nastavit. Protože ukazatel jsme používali pro ukládání dat, je nyní nastaven na konec oblasti uložených dat. Právě proto musíme jeho hodnotu nastavit tak, aby ukazoval na začátek oblasti dat v paměti RAM. Toto provedeme dvěmi instrukcemi typu LDI a s pomocí funkcí LOW a HIGH, stejně jako na začátku celého programu.

Příprava na vysílání synchronizačních bajtů

Ukazuje se, že při vysílání dat do počítače je nutné vložit mezi vysílaná data jedinečné synchronizační bajty. Důvody jsou dva. Je nutné, aby nedošlo k prohození pořadí bajtů, tedy aby nedošlo k záměně horního a dolního bajtu jednoho vzorku. Druhým důvodem je oddělení jednotlivých sekvencí 401 vzorků, protože kdybychom pomocí počítače změnili vzorkovací rychlost, tak bychom na obrazovce pravděpodobně obdrželi první část oscilogramu v jedné vzorkovací rychlosti a druhou část už ve vzorkovací rychlosti jiné, nově nastavené. Synchronizační bajty nám významně pomohou udržet pořádek v systému dat.

Než cokoli pošleme pomocí jednotky UART, musíme se ujistit, je-li vysílací buffer prázdný. Volnost tohoto bufferu nám indikuje bit UDRE v registru UCSRA.

Vysílané sekvence 401 vzorků (802 bajtů) jsou odděleny dvěma synchronizačními bajty, každý o desítkové hodnotě 128. Každé dva posílané bajty jsou pak odděleny jedním bajtem o hodnotě 128. Z předchozích opatřeních vyplývá, že hodnotu 128 nikdy nenabude žádný datový bajt.

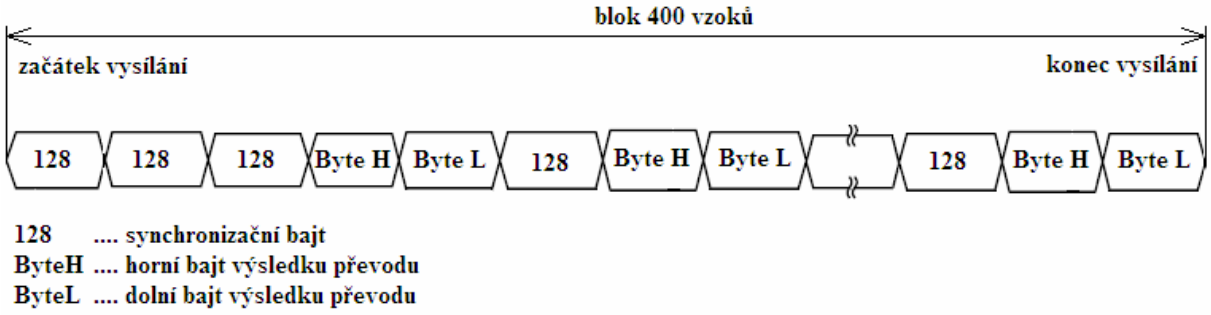
Instrukcí SBIS otestujeme stav bitu UDRE. Je-li tento bit v logické nule, pak vysílací buffer není prázdný a použitá přeskoková instrukce v této situaci nebude nijak měnit tok programu. V tomto případě bude vykonána hned následující instrukce, tedy instrukce relativního skoku RJMP, která program vrátí zpět na návěští SYNCHROBAJT1. Bude-li bit UDRE v logické jedničce, pak je buffer prázdný a instrukce SBIS způsobí přeskočení instrukce RJMP a vykonána bude až instrukce OUT. Touto instrukcí zkopírujeme obsah registru R20 do datového registru UDR. Mikrokontrolér pak automaticky pošle požadovanou hodnotu sériovou linkou. V registru R20 je už od samého začátku kódu hodnota 128.

Stejným způsobem je odeslán i druhý synchronizační bajt o stejné hodnotě. Odeslání těchto dvou bajtů je signálem pro program v počítači, aby následující přijaté bajty považoval za blok 401 vzorků, přičemž je každý vzorek oddělen dalším synchronizačním bajtem.

Načítání naměřených dat z paměti RAM

Načtení uložených vzorků z vnitřní datové paměti RAM je provedeno instrukcí LD. Vždy byl nejdříve uložen horní bajt vzorku a po něm i bajt dolní. Proto musíme data načítat ve stejném pořadí tak, jak byla uložena. Při načítání opět použijeme ukazatel, který již máme správně nastaven na začátek uložených dat. V kombinaci s ukazatelem používáme i inkrementační operátor, abychom nemuseli provádět inkrementování ukazatele samostatnou instrukcí. Horní bajt vzorku načteme do R22, dolní pak do R23.

Jak již bylo řečeno, je před každým vyslaným vzorkem vyslán jeden synchronizační bajt. Takže na úplném začátku přenosu jsou vyslány tři synchronizační bajty, pak dva bajty dat, dále následuje jeden synchronizační bajt, dva datové, atd. Konec jednoho vysílaného bloku 401 vzorků a začátek dalšího bloku, vypadá následovně: jeden synchronizační bajt, dva datové, tři synchronizační, dva datové. Vysílání jednoho bloku 400 (401) vzorků do počítače je znázorněno na obr. 28. Synchronizační bajt, oddělující jednotlivé vzorky má hodnotu 128, je tedy identický s předchozími synchronizačními, označující začátek bloku 400 vzorků dat.



Obr. 28 Relace vysílání naměřených dat do počítače

Vysílání načtených vzorků do počítače

Horní bajt naměřeného vzorku je vysílán od návěští HBYTE. Prozatím je tento bajt uložen v registru R23. Do vysílacího registru nemůžeme vkládat data, dokud nebude bit UDRE nastaven do logické jedničky. K testování tohoto bitu v registru UCSRA opět použijeme instrukci SBIS. Dokud nebude UDRE ve stavu logické jedničky, bude program neustále vracen na testování tohoto bitu instrukcí RJMP.

V opačném případě je instrukcí OUT zapsán obsah R23 do datového registru UDR jednotky UART. Ihned po odvysílání horního bajtu je stejným způsobem odeslán i dolní bajt. Tento bajt je odesílán od návěští LBYTE.

Odpočítání vyslání 401 vzorků

Operace počítání odeslaných vzorků je prováděna úplně stejně jako při načítání dat do RAM. Nejdříve je dekrementován první počítací registr R17, pokud jeho hodnota nebude nula, pak se program vrátí na návěští NACTI, kde bude z RAM načten další vzorek. Pakliže jeho hodnota bude nula, tak pomocí podmíněného skoku BREQ bude tok programu přesunut na návěští DALSI_DEC1. Zde bude registru R17 přiřazena hodnota jedna, aby se program od téhle chvíle až po načtení dalšího vzorku stále vracel na toto návěští. Dále je dekrementován i druhý počítací registr R18. Jeho hodnota je pak testována přeskokovou instrukcí CPSE, tak že je porovnávána s nulovou hodnotu v registru R19. Pokud tedy R18 ještě nebude nulový, tj. nebude dosud odesláno 401 vzorků, bude program vrácen na návěští NACTI pomocí instrukce RJMP. Při načtení a odeslání potřebného počtu vzorků je hlavní smyčka programu vykonaná. Pak už nezbývá nic jiného, než pomocí instrukce pro relativní nepodmíněný skok RJMP zakončit smyčku, převedením toku programu na návěští CEKEJ_PRIKAZ, jenž je samým začátkem hlavní smyčky.

Rutina pro časování odběru vzorků CEKEJ

Pro časování by se mohlo použít některého ze tří integrovaných časovačů v mikrokontroléru. Jednodušší se však ukázalo časování zdržením průchodu programu nějakými instrukcemi, přičemž víme, že jeden instrukční cyklus je vykonáván 62,5 ns a známe i přesný počet instrukčních cyklů pro vykonání jednotlivých instrukcí. Vzorkovací rychlost použitého A/D převodníku je 100 kSPS. Čas mezi odběrem sousedních vzorků je tedy 10 μs. Rutina CEKEJ musí mít v základním režimu délku 10 μs, zmenšenou o čas, který je potřebný na vykonávání předchozích instrukcí. Abychom byli schopni snížit vzorkovací rychlost na polovinu, tedy abychom zajistili dvojnásobné zvětšení doby mezi odběry vzorků, musíme do rutiny zavést nějakou proměnnou, jejíž hodnotu budeme měnit prostřednictvím povelů z počítače.

Bylo spočteno, že rutina musí pozdržet běh programu minimálně o 8,125 μs. Této době odpovídá asi 130 instrukčních cyklů. Musíme však vzít do úvahy i pomocné instrukce. Proto je do registru R16 načtena hodnota 26. Tato hodnota je následně dekrementována a je testována na nulovou hodnotu pomocí instrukce podmíněného skoku BRNE.

Dokud tedy hodnota v R16 není nulová, je program vracen zpět na dekrementační instrukci označenou návěštím A. Až bude R16 mít nulovou hodnotu, bude dekrementován registr R29, který slouží jako nastavovací proměnná. V základním režimu, tedy při vzorkování maximální rychlostí, je jeho hodnota právě jedna. Bude-li totiž jeho hodnota po dekrementování nulová, bude se jedná o vzorkování maximální rychlostí a nebude třeba dále zpožďovat běh programu a instrukcí RET bude tok programu vrácen na místo, odkud byl zavolán. Pro maximální rychlost je tato smyčka tedy proběhnuta jedenkrát. Pokud budeme chtít vzorkovat například čtvrtinovou rychlostí, tedy 25 kSPS, musíme zajistit, aby rutina pozdržela běh programu čtyřikrát déle. Toho docílíme vložením hodnoty 4 do registru R29. Po průběhu první částí rutiny CEKEJ, která i s časem na provedení instrukcí v hlavní smyčce zpomalí program asi 10 μs, je nutné zajistit, aby byla tato první část provedena ještě třikrát. V R29 máme hodnotu 4, tato hodnota je dekrementována. Instrukce BRNE zjistí, že R29 nemá nulovou hodnotu a vrátí program zpět na začátek rutiny CEKEJ. Tady znovu proběhne zdržení téměř o 10 μs, pak je hodnota v R29 opět snížena o jedničku a testována na nulovou hodnotu. Program bude tedy celkem třikrát vrácen na začátek čekací rutiny, čímž dosáhneme zpoždění: 10 μs + 3 * 10 μs = 40 μs. To odpovídá požadované vzorkovací rychlosti 25 kSPS.

4.2 Program pro počítač

Software je také velice důležitou součástí popisovaného zařízení. Byl zvolen programovací jazyk MICROSOFT VISUAL C++ 6.0 – výuková verze, distribuovaná zdarma. Zdrojové kódy všech souborů jsou umístěny na doprovodném CD.

Z důvodů vysoké rozsáhlosti všech potřebných souborů (asi 40 stran A4) není jejich celkový popis možný. Pro tvorbu tohoto programu nám ani obsah všech potřebných soborů nemusí být zcela znám. VISUAL C++ si totiž, mimo pomocných kompilačních souborů, sám vytvoří i další pomocné hlavičkové i jiné „zdrojové“ soubory. Například, pokud si vytvoříme vlastní ikonu v grafickém prostředí VISUAL, pak si tento program sám vytvoří soubor resource.h. V takovém případě se o obsah takového souboru nemusíme zajímat. Máme také možnost použit různých průvodců, pak je automaticky vytvořeno velké množství souborů, v jejichž systému můžeme ztratit přehled. Proto bude dále popsána pouze struktura zdrojového souboru HlavniOkno.cpp, který obsahuje vytvoření hlavního okna aplikace, ovládacích prvků a dále definice důležitých funkcí pro zpracovaní dat, deklarace a definice dalších pomocných proměnných a další součásti. V tomto souboru jsou implementovány hlavní funkční algoritmy pro zpracování dat, naměřených A/D převodníkem.

Ke konvertoru USB/UART je k dispozici hlavičkový soubor Ftd2xx.h, ve kterém jsou nadefinovány možné přenosové rychlosti, time-out mezi vysíláním a také prototypy funkcí, s jejichž pomocí lze na port USB zapisovat a také ho číst. Od operačního systému Windows 98 už nelze jednoduše přistupovat na porty. Je nutné mít vytvořené speciální funkce, které nám toto umožní. Bližší informace např. v [6].

Soubor Pripojeni_USB.h je hlavičkovým souborem, ve kterém je vytvořena třída CPripojeni_USB, jenž obsahuje prototypy funkcí a deklarace globálních proměnných, které budou používány v hlavním programu. Zdrojový soubor Pripojeni_USB.cpp obsahuje konstruktor a destruktor třídy CPripojeni_USB a také definice všech funkcí, deklarovaných v stejnojmenném hlavičkovém souboru. Hlavičkový soubor PrvniApp.h obsahuje metody, s jejichž pomocí je celá aplikace inicializována. Je zde vytvořena třída CPrvniApp, která je odvozena od standardní třídy CWinApp, jenž slouží pro vytvoření hlavního okna výsledné aplikace. Zdrojový soubor PrvniApp.cpp obsahuje konstruktor a destruktor třídy CPrvniApp. V tomto souboru je definice inicializační a ukončovací metody hlavního okna aplikace. Hlavičkový soubor HlavniOkno.h obsahuje zavedení třídy CHlavniOkno, která obsahuje prototypy, tedy deklarace všech použitých funkcí, deklarace ovládacích prvků a proměnných.

Popis hlavního souboru HlavniOkno.cpp

Na začátku kódu jsou do programu pomocí direktivy pro vkládání načteny pomocné soubory. Definičními direktivami jsou ke všem použitým ovládacím prvkům přiřazeny jedinečné hodnoty, tzv. ID identifikátory. Ovládacími prvky jsou například tlačítka, textová pole či výklopný seznam – combo box.

Následuje oblast mapy zpráv, ve které jsou systémovým zprávám přiřazeny obslužné funkce, spolu s příslušným ovládacím prvkem. Dále jsou definovány další globální proměnné. Například pro navázání komunikace s FT232BM jsou důležité tyto proměnné: verifikace, ftHandle, status, pUSB a asituace. Pro práci s naměřenými daty jsou podstatné tyto proměnné: vzorkovani, zacatek_x_osy, zacatek_y_osy a především dataAD[401], což je jednorozměrné pole, které slouží pro uložení přijatých napěťových vzorků. Toto pole je nutné inicializovat, tedy každému prvku pole přiřadit nějakou počáteční hodnotu, nejlépe nulu. Pro tuto inicializaci je vytvořená funkce Inicializace_dataAD.

Dále je definována třída hlavního okna aplikace, je definována jeho velikost, možnost minimalizace a jeho název. V tomto místě jsou také vytvořeny a inicializovány veškeré ovládací prvky. Jsou zde vytvořeny oddělovací rámečky, textové pole pro zobrazovaní úspěšnosti zahájené komunikace, veškerá tlačítka, výklopný seznam pro zvolení rychlosti vzorkování, obrázek jako podkladový rastr pro měření a další prvky. Pro výklopný seznam je nutné nadefinovat, co se vlastně má v seznamu ukazovat a tyto zobrazované symboly je nutné kvalifikovat – přiřadit jím určité hodnoty.

Pro zjednodušení ovládání je použita funkce DeaktivaceTlacitek, která způsobí optické zašednutí všech tlačítek, které se nebudou účastnit procesu zahájení komunikace. Obsluha pak nebude mít na výběr z několika tlačítek, protože v aktivním režimu bude pouze tlačítko pro pokus o navázání spojení s připojeným zařízením.

Další částí je destruktor třídy CHlavniOkno, jenž má za úkol uvolnit použité systémové prostředky. Následuje definice obslužných funkcí pro tlačítka KONEC a HLEDEJ_HW. Funkce OnHledej_KIT() je zavolána po stisku tlačítka HLEDEJ_HW. Tato funkce má za úkol navázat spojení s připojeným zařízením. Obsahuje i systém zpráv uživateli o úspěšnosti či neúspěšnosti připojení zařízení k počítači. Funkce Vyber_proveden() slouží pro výběr zařízení v případě, že je připojeno více stejných nebo podobných zařízení. Obsahuje též funkce pro zahájení a ukončení komunikace přes USB.

OnPaint je standardní funkcí, která je volána při překreslování údajů na obrazovce. Na jejím začátku je získán tzv. kontext zobrazovacího zařízení a to typu dc. Na obrazovku jsou pak vypsány statické textové popisky. Na konci funkce je volána funkce pro zobrazovaní dat. To proto, že při minimalizaci okna aplikace nebo při ztrátě tzv. fokusu by došlo ke smazání vykresleného průběhu. Aby nedocházelo k vykreslování křivky při navázání komunikace, je v této funkci použito jednoduchého testu, který ověřuje, kdy byla OnPaint() zavolána.

Následuje definice již použité funkce ProcessMessages. V definici funkce OtevriProPrenos je nastavena přenosová rychlost, a sice na hodnotu 115200 Bd. Stejná rychlost musí být nastavena i v programu pro mikrokontrolér, jinak nebude komunikace vůbec pracovat. Jsou zde nastaveny také hodnoty pro time-out při komunikaci.

Stěžejní funkce pro práci s daty je pojmenovaná Nacti_data. Jejím úkolem je vyslání požadavku na načtení bloku 401 vzorků. Požadavek přebírá FT232BM a postupuje ho mikrokontroléru. Požadavek je ve formě osmibitového čísla. Toto číslo symbolizuje požadovanou rychlost vzorkování. Tato funkce obsahuje hlavní cyklus (DO – WHILE). Bude tedy probíhat, dokud bude splněna určitá podmínka. Touto podmínkou je v konečném důsledku stisknutí tlačítka STOP v ovládacím okně programu. Program ve smyčce nejdříve čeká na příchod dvou synchronizačních bajtů, jejichž číselná hodnota je 128 (viz program pro mikrokontrolér). V mikrokontroléru je zaručeno, že žádný jiný bajt nenabude této hodnoty. Jsou-li zaznamenány dva synchronizační bajty, pak program pokračuje na načítání naměřených vzorků po bajtech. Toto načítání je řízeno pomocí cyklu FOR. Jeden vzorek je složen ze dvou bajtů. Aby bylo zaručeno, že nedojde k záměně horního a dolního bajtu jednoho vzorku, je před příjem každého vzorku zařazen cyklus, který čeká na příchod dalšího synchronizačního bajtu. Jeho zaznamenání tedy signalizuje příchod nového vzorku. Po synchronizačním bajtu je z USB přečtena hodnota horního bajtu a uložena do proměnné dataH. Dolní bajt je načten jako druhý a uložen do proměnné dataL. Nyní je třeba tyto dva bajty převést na měřené napětí. Tento převod provádějí funkce Spocti_napeti_Kl() a Spocti_napeti_Zp(). První převádí kladné hodnoty měřeného napětí, druhá záporné. Proto je nejdříve nutné rozhodnout, zda naměřený vzorek představuje napětí kladné či záporné. Toto je prováděno pomocí testování nejvyššího bitu horního bajtu výsledku. Tuto možnost nám poskytuje výstup dat z A/D převodníku ve formě dvojkového doplňku. Pakliže bude mít MSB hodnotu nula, tak naměřený vzorek odpovídá kladnému napětí. Bude-li jeho hodnota jedničková, jedná se o napětí záporné.

Funkce Spocti_napeti_Kl() nebo Spocti_napeti_Zp() jsou volány dvakrát Nejdříve pro převod horního bajtu a po druhé pro převod dolního. Pomocí jejich parametrů se definuje o jaký bajt se jedná, jestli horní nebo dolní. Výsledky převodu horního i dolního bajtu vzorku jsou sečteny a uloženy do matice dataAD. Jeden ze 401 vzorků je tedy načten, převeden a uložen. Program se ve smyčce FOR vrací na odběr dalšího vzorku, tedy načtení synchronizačního bajtu a dvou bajtů dalšího výsledku měření. Je-li načteno 401 vzorků je cyklus FOR ukončen a je zavolána další funkce Zobraz_data.

Tato funkce provádí vlastní zobrazování naměřených dat na obrazovku počítače a proto bude popsána samostatně v jiném odstavci. Po zápisu na USB ve funkci Nacti_data() je ještě zařazen test, kterým tlačítkem byla tato funkce zavolána. Standardně je volána po stisku tlačítka START. Může ji však volat i tlačítko reset, které slouží pro vynulování parametrů zobrazovaných dat a dat samotných. Takový parametr je např. offset napětí, s jehož pomocí lze posouvat měřených průběh vertikálně po obrazovce. Pokud bylo stisknuto tlačítko reset, pak je vynechána fáze načítání vzorků, matice s převedenými vzorky je pomocí speciální funkce vymazána, jsou vynulovány offset parametry a funkce Zobraz_data() je zavolána se změněným parametrem, který slouží pro oddělení zobrazení měřeného napětí a stavu po resetu.

Další částí programu je definice obslužných funkcí pro jednotlivá tlačítka. První je definována funkce pro tlačítko START. Zde je vytvořena proměnná A, jenž je přiřazena hodnota 1. Globální proměnné konec je přiřazena hodnota false, což slouží jako podmínka pro hlavní smyčku funkce Nacti_data(). Nakonec je zavolána funkce Nacti_data(), přičemž hodnota proměnné A je této funkci předána hodnotou. Stisk tohoto tlačítka vyvolá standardní práci funkce Nacti_data().

Tlačítko reset je definováno podobně jako předchozí. Lokální proměnné *povel* je přiřazena hodnota 36, což ve funkci Nacti_data() způsobí výše popisovaný reset všech parametrů a naměřených hodnot. Globální proměnné *konec* je přiřazena hodnota true, což způsobí, že funkce Nacti_data() je proběhnuta pouze jednou. Nakonec je tato funkce volána.

Následují čtyři tlačítka pro posun naměřeného průběhu nahoru, dolů, vlevo či vpravo. Všechna jsou definována podobně. Tlačítko pro posun průběhu doprava při svém stisku zvýší hodnotu parametru zacatek_x_osy o jedničku a zavolá se funkce Zobraz_data(). Tlačítko posunu vlevo parametr o jedničku snižuje. Posun ve vertikálním směru provádí tlačítka UP a DOWN. Pracují stejně jako předchozí dvě, pouze mění parametr zacatek_y_osy a to přičtením nebo odečtením hodnoty 5 při jejich stisknutí.

Dalšími funkcemi jsou DeaktivaceTlacitek() a AktivaceTlacitek(). První funkce po jejím zavolání způsobí zašednutí všech tlačítek, které jsou vypsány v jejím těle. Toto slouží pro snadnější orientaci při používání programu. Na začátku ovládání nemá totiž uživatel možnost výběru z více tlačítek, a proto je alespoň jeho spouštění snadnější. Druhá funkce všechna tlačítka zpřístupní k ovládání.

Dále je provedena definice funkcí pro převod naměřených bajtů na desítkovou hodnotu měřeného napětí. Pracují na principu testování jednotlivých bitů naměřených bajtů, přičemž jsou bitům přiřazeny napěťové hodnoty dle předávaných parametrů. Pokud například chceme převést jeden kladný vzorek napětí, tedy dva bajty, pak je nejdříve zavolána funkce v tomto tvaru: Spocti_napeti_Kl(& a5, & dataH, 2, 64). Parametr a5, předávaný ukazatelem slouží pro dočasné uložení převedeného napětí, dataH je ukazatel na právě převáděný bajt, parametr j = 2 přiděluje prvnímu testovanému bitu hodnotu 2,5 / j = 2,5 / 2 = 1,25 V. Parametr i = 64 říká, že testování bajtu začne až o druhého nejvyššího (2^6 = 64), protože první je znaménkový. Po dokončení převodu prvního – horního bajtu se zavolá funkce znovu pro bajt dolní takto: Spocti_napeti_Kl(& a5, & dataL, 256, 128). První parametr je stejný, druhý nám říká, že převádíme dolní bajt, hodnota 256 určuje, že nejvyššímu testovanému bitu odpovídá hodnota 2,5 V / 256 = 0,0098 V. Parametr 128 sděluje funkci, aby testování prováděla od nejvyššího – sedmého bitu (2^7 = 128).

Vykreslování křivek naměřeného napětí na obrazovku je prováděna pomocí funkce Zobraz_data. Na začátku kódu této funkce je zavolána funkce OnPaint, jenž musí být volána při každém překreslení obrazovky. Je vytvořen speciální font pro zobrazování číselných hodnot napětí. Jsou také založena dvě pera pro kreslení. Červené pro vykreslování měřeného napětí a modré, sloužící pro rozlišení stavu resetu od měření napětí s velmi nízkou amplitudou. Dále je získán kontext zobrazovacího zařízení typu dc i *pDC. V tomto kontextu je ještě nastaveno průhledné pozadí pro vytvořený font. Jako pozadí pro zobrazení měřeného průběhu je vybrána bitmapa, uložená v proměnné obrazek. Dále je zařazen test, jestli před voláním této funkce bylo stisknuto tlačítko reset. Pokud ano, pak se barva kreslícího pera změní na modrou a bude znamenat nulové napětí.

Nyní je vše připraveno pro vykreslení průběhu měřeného napětí. Vykreslení je prováděno v kontextu pDC. Vlastní kreslení je prováděno pomocí funkcí MoveTo a LineTo. První funkce nastaví pozici kurzoru na zadané souřadnice, druhá spojí aktuální pozici s novou pozicí, zadanou dvěmi souřadnicemi. Nejdříve je tedy pozice kurzoru nastavena na hodnotu prvního, resp. nultého vzorku napětí z matice dataAD.

Spojení všech 401 vzorků pomocí funkce LineTo je realizováno cyklem FOR. Při každém průchodu cyklu je navíc testována pozice každého měřeného napěťového bodu vůči rastru a také jsou zjišťovány extrémní hodnoty napětí, pro výpočet maximálního napětí a napětí špička – špička. V cyklu je také vypočítávána průměrná a efektivní hodnota vstupního napětí.

Pokud by pozice křivky, představující měřené napětí, nebyla sledována, pak např. při vertikálním posunu průběhu pomocí tlačítka DOWN by jinak došlo k vykreslování průběhu do oblasti ovládacích tlačítek, což je nežádoucí. Podle pozice průběhu jsou proto jednotlivá tlačítka zakazována nebo povolována, jestli jsou v danou situaci smysluplná. Nakonec jsou vytvořené instance použitých tříd zrušeny (fonty a pera).

Jako poslední jsou uvedeny definice funkcí pro obsluhu zaškrtávacích políček, combo boxu a inicializační funkce pro matici dataAD. Při zaškrtnutí políčka „1:10“ jsou zobrazované parametry vynásobeny číslem 10, zobrazovaná hodnota počtu voltů na dílek rastru je také desetkrát zvětšena. Obsluha výklopného seznamu pracuje tak, že se zjistí jeho zvolená položka, z té se vypočte vzorkovací rychlost, která je následně zobrazena. Nakonec obslužné funkce je zavolána funkce Nacti_data(), kterou se zajistí odběr vzorků. Protože však v této volané funkci nebude splněna podmínka hlavního cyklu, dojde k odběru pouze jednoho bloku 401 vzorků, program tedy bude po tomto načtení a zobrazení dat zastaven, aby uživatel měl možnost změnit nastavovanou vzorkovací rychlost ve výklopném seznamu, pokud mu tato nově nastavená nevyhovuje. Pokud je vše v pořádku, stiskem tlačítka START je možné program rozběhnout v cyklu.

5. Popis funkce zařízení

5.1 Softwarové ovládání zařízení

Nejdříve je nutné připojit zařízení na sběrnici USB k počítači. Při prvním připojením je nutné nainstalovat ovladač, dodávaný k obvodu FT232BM. Je také možné naprogramovat paměť EEPROM a vložit do ní název zařízení nebo identifikátor. Pak pomocí speciální aplikace naprogramujeme mikrokontrolér a resetujeme zařízení oběma tlačítky nebo odpojíme a znovu připojíme zařízení k PC.

Konečně je možné spustit ovládací program, jehož hlavním okně bude prozatím k dispozici jen jediné funkční tlačítko – NAJDI HW. Stiskem tohoto tlačítka bude navázána komunikace se zařízením. Pokud dojde k úspěšnému rozeznání připojeného zařízení, pak bude v textovém poli zobrazen stav navázání komunikace. Ve výběrovém textovém poli bude zobrazen název připojeného zařízení, který je uložen v paměti EEPROM. Na tento zobrazený název je nutné klepnout myší. Tím se nám zpřístupní dosud nepřístupná tlačítka a na rastru je zobrazena modrá přímka.

Pro ovládání zařízení máme k dispozici tlačítka pro spuštění měření (START), zastavení měření (STOP), posunu měřeného průběhu po obrazovce (šipky) a pro resetování naměřených dat (RESET). Pro změnu vzorkovací rychlosti slouží výsuvný seznam (Combo Box). Pokud budeme chtít práci programu ukončit, pak stiskneme tlačítko KONEC.

Vedle výklopného seznamu, kterým volíme rychlost vzorkování v podobě časového intervalu na dílek rastru je umístěn popisek, který nám ukazuje nastavenou vzorkovací rychlost v jednotkách, ve kterých je to zvykem, tedy v kSPS či SPS (Samples Per Second). Maximálně lze nastavit 200 μs/dílek, což odpovídá vzorkovací rychlosti 100 kSPS, mělo by tedy dojít k odběru 100 000 vzorků za sekundu. Jelikož je akviziční paměť pouze pro 401 vzorků, tak je odebráno 401 vzorků, ale s periodou, odpovídající rychlosti vzorkování.

V okně ovládacího programu je standardně zobrazena jen průměrná hodnota měřeného napětí. Zaškrtávací políčko „AC“ slouží pro zobrazení i dalších měřených parametrů, které pro stejnosměrná měření nemají valného smyslu. Těmito dalšími parametry jsou: napětí „špička – špička“, maximální napětí a efektivní hodnota napětí. Pokud budeme měřit neměnné stejnosměrné napětí, pak políčko „AC“ necháme prázdné. Na obrazovce uvidíme měřený signál, zobrazený červeně a jeho střední hodnotu, zobrazenou černě.

Pokud chceme sledovat střídavé signály, pak zaškrtneme políčko „AC“ a na obrazovce budeme mít výsledný tvar měřeného signálu a čtyři výše zmíněné charakteristiky jeho amplitudy, křivka měřeného napětí a zobrazené číselné charakteristiky budou zobrazeny v barvě červené, pro odlišení od stejnosměrných měření.

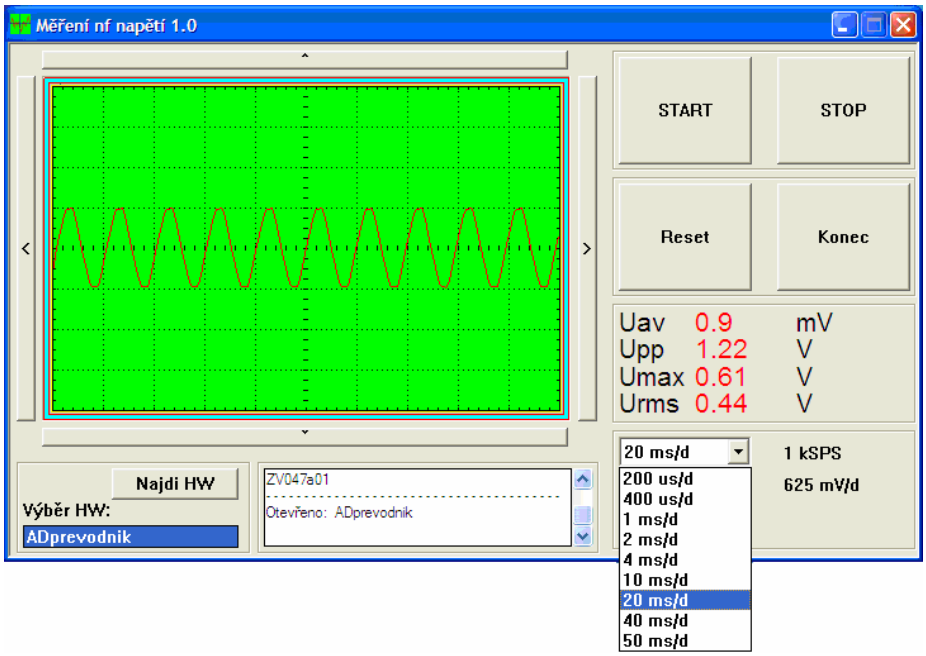
Zaškrtávací políčko „1:10“ slouží pro změnu vertikálního měřítka na obrazovce. Zařízení však nemá možnost měnit zesílení signálu, proto je toto políčko využitelné jen při připojení signálu přes dělič 10:1. Při zaškrtnutí tohoto políčka budou naměřené hodnot vynásobeny číslem 10. Hodnota, představující počet voltů na dílek bude také automaticky desetkrát zvětšena.

Pro start měření je (překvapivě) nutné stisknout tlačítko START. Na obrazovce se ihned začne zobrazovat průběh vstupního měřeného napětí. Výsledný průběh představuje červená křivka o tloušťce jeden pixel. Vedle průběhu bude zobrazována také jeho střední hodnota. Pokud bude zaškrtnuto políčko „AC“, pak budou zobrazeny i další hodnoty. Vedle výklopného seznamu bude zobrazena nastavená vzorkovací rychlost. Pod tímto údajem nalezneme také hodnotu, která představuje počet voltů na dílek rastru, abychom si případně mohli změřit určité další napěťové parametry signálu. Výklopný seznam ukazuje hodnotu časového údaje na dílek rastru pro zjištění periody a potažmo i frekvence měřeného signálu.

Jelikož zařízení nedisponuje žádným synchronizačním obvodem, výsledný průběh se pak po obrazovce horizontálně posouvá a to tím pomaleji, čím nižší vzorkovací rychlost je nastavena. Pokud máme nastavenou vysokou vzorkovací rychlost pro relativně pomalý průběh, pak může být průběh zobrazován zbytečně trhaně, proto je u tohoto zařízení vhodnější volit vždy spíše menší rychlost vzorkování a mít na obrazovce několik period měřeného signálu. Tento zobrazovaný průběh bude automaticky obnovován. Tlačítkem STOP můžeme kdykoli měření zastavit. Aby se zvýšila jednoduchost ovládání, tak jsou při stisknutí tlačítka START zakázána ty tlačítka, jejichž použití není v danou situaci vhodné. Tedy například při započatém měření není vhodné měnit vzorkovací rychlost nebo zavírat okno aplikace nebo se pokoušet spustit spuštěné měření.

Pokud požadujeme změnu vzorkovací rychlosti, pak je nejdříve nutné načítání dat do počítače zastavit tlačítkem STOP a poté stisknout šipku u výklopného seznamu (jinou možnost ani v tuto chvíli nemáme). Po jejím stisknutí dojde k rozbalení seznamu, ve kterém máme na výběr z několika vzorkovacích rychlostí, které jsou však určeny pomocí časového údaje na dílek rastru obrazovky. Nastavování vzorkovací rychlosti je tedy stejné jako u digitálních osciloskopů.

Ze seznamu vybereme požadovanou rychlost tak, že na ní klepneme pomocí myši. Seznam se automaticky zabalí a dojde k odběru jednoho bloku 401 vzorků, čímž se na obrazovce zobrazí průběh měřeného napětí, vzorkovaného nově nastavenou rychlostí. Hodnota vzorkovací rychlosti v jednotkách SPS nebo kSPS bude také změněna po zavření výklopného seznamu. Pokud nám nově nastavená vzorkovací rychlost vyhovuje, pak stiskem tlačítka START dojde ke startu měření vstupního napětí požadovanou rychlostí vzorkování. Obr. 29 ukazuje, jak lze vzorkovací rychlost měnit.



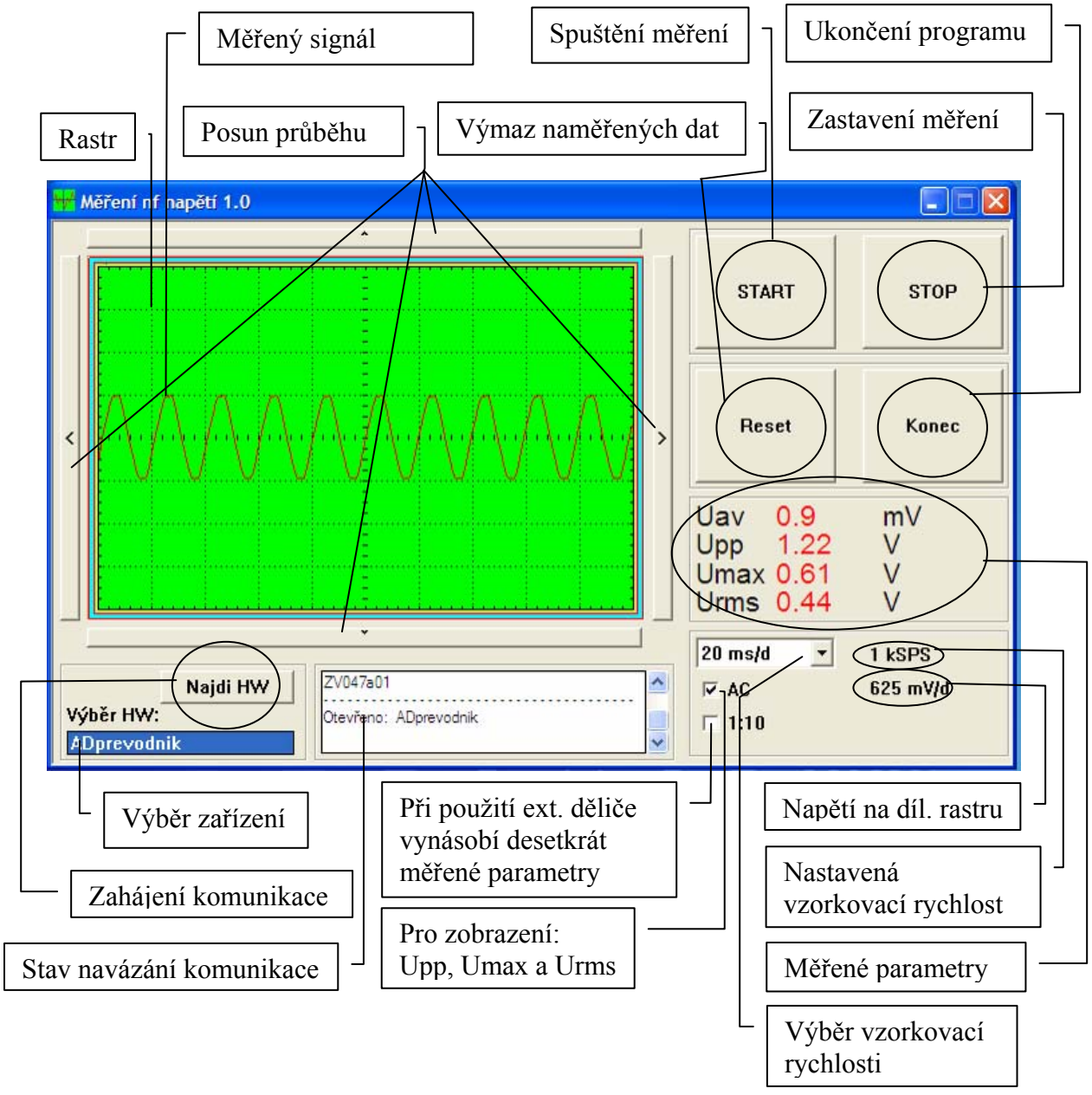
Obr. 29 Způsob změny vzorkovací rychlosti

Ukončení měření provedeme opět tlačítkem STOP. Teprve až po jeho stisknutí máme možnost resetovat měřené hodnoty, dále měnit vzorkovací rychlost nebo ukončit práci aplikace pomocí tlačítka KONEC.

Tlačítko RESET slouží pro vynulování naměřených dat a parametrů, kterými můžeme měnit pozici průběhu měřeného signálu na obrazovce. Po jeho stisknutí bude na stínítku zobrazena silnější modrá přímka, ležící vodorovně uprostřed rastru, na kterém byl předtím zobrazován průběh měřeného napětí. Modrá barva přímky je zvolena pro rozlišení stavu po resetu od stavu měření napětí s nulovou amplitudou.

Poslední možností pro práci s naměřenými daty jsou čtyři tlačítka, sloužící pro posun celého naměřeného průběhu nahoru, dolů, vlevo či vpravo na měřícím rastru obrazovky. Pokud totiž chceme měřit například amplitudu vstupního napětí pomocí rastru, pak si můžeme snadno posunout průběh tak, abychom mohli snadněji odečíst požadovanou hodnotu.

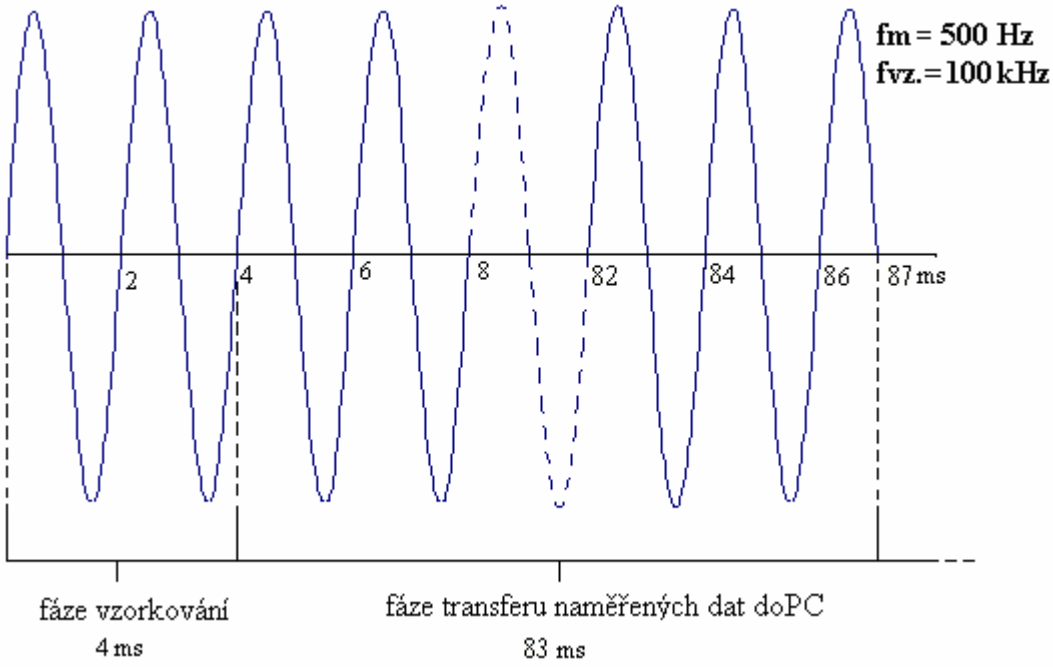
Celkový počet posunutí ve vodorovném směru je limitován. Dojde-li k překročení povoleného počtu posunutí v jednom směru, pak bude příslušné tlačítko zakázáno, nebude tedy dále možné ho použít a místo parametrů měřeného napětí bude zobrazeno „NO POSTDATA“ nebo „NO PREDATA“, dle směru posunu. Při posouvání průběhu mimo rastr dojde na jednom konci zobrazeného průběhu ke vratné ztrátě části křivky, na druhé straně bude průběh doplňován křivkou nulového napětí. Při posouvání ve vertikálním směru dojde jen k zakázání toho tlačítka, jehož stisknutí by vedlo k vykreslování průběhu mimo rastr. Na obr. 30 je zobrazen ovládací program „v akci“ a popis jeho ovládacích prvků.



Obr. 30 Popis ovládacích prvků programu (měřený signál s frekvencí 50 Hz)

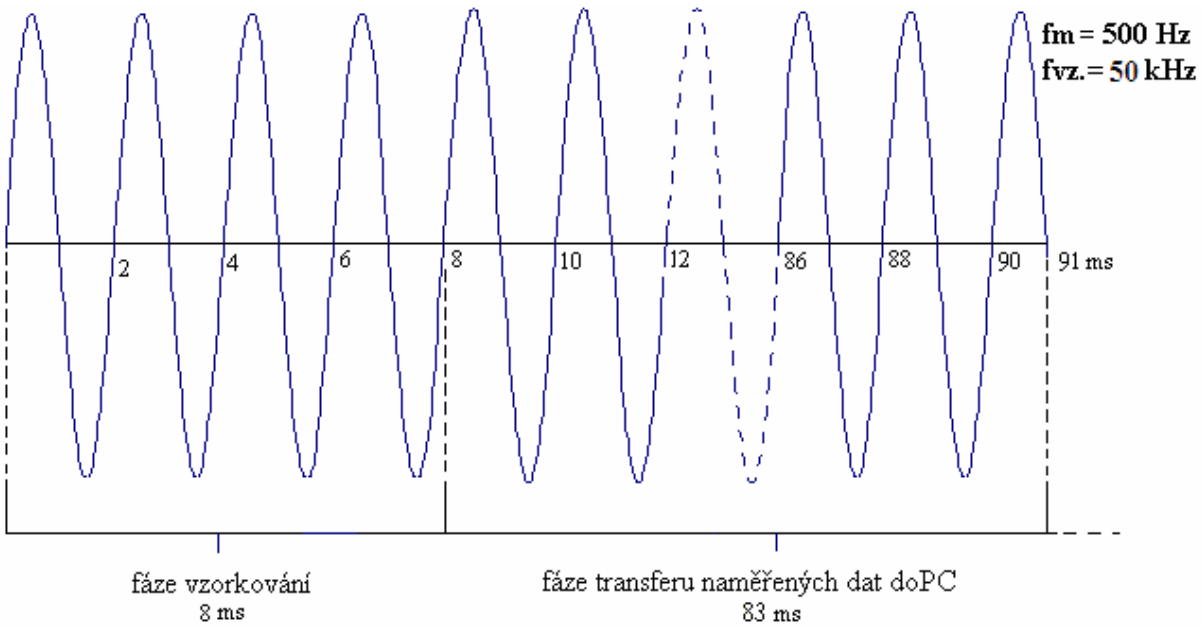
5.2 Rozbor měření pomocí zařízení

Zařízení pracuje s maximální rychlostí vzorkování 100 kSPS (Real Time). Měřený signál však není sledován neustále, nýbrž v periodických intervalech, dle nastavené rychlosti vzorkování. Pokud například chceme změřit časový průběh sinusového signálu s kmitočtem fm = 500 Hz pomocí popisovaného zařízení, ve kterém jsme nastavili vzorkovací rychlost na fvz = 100 kSPS, pak uvažujeme takto: Perioda vzorkování je 10 µs. Z měřeného signálu je odebráno 400 vzorků, které jsou od sebe vzdáleny 10 µs. Doba, po kterou je měřený signál sledován a vzorkován je proto: 400 * 10 µs = 4 ms. Po tuto dobu je sledován každý měřený signál, který je vzorkován rychlostí 100 kSPS. Měřený signál s kmitočtem fm = 500 Hz je tedy také sledován po dobu 4 ms, tedy po dobu jeho dvou period (Tm = 2 ms). Po získání a konverzi všech napěťových vzorků jsou naměřená data odeslána do počítače. Přenosová rychlost je zvolena 115,2 kBd (115,2 kb/s), tj. 14,4 kB/s (115,2 kBd / 8 b). Jeden vzorek je složen z horního a dolního bajtu a opatřen synchronizačním bajtem. Blok 400 vzorků tedy představuje: 400 * 3 = 1,2 kB. Čas, který potřebujeme na odeslání naměřených dat do počítače je tedy: 1,2 / 14,4 ≈ 83 ms. Tato doba je konstantní pro všechna měření a nezávislá na nastavené vzorkovací rychlosti. Celou situaci měření sinusového signálu s kmitočtem 500 Hz, pomocí vzorkovacího signálu 100 kSPS, je vidět na obr. 31. Tam je zobrazen průběh vstupního signálu, rozděleného na část vzorkovací, kdy je zařízením sledován a část, kdy je prováděn přenos dat do počítače.



Obr. 31 Měření sinusového signálu 500 Hz pomocí vzorkovací rychlosti 100 kSPS

Pro lepší pochopení principu funkce popisovaného zařízení následuje krátký rozbor dalšího příkladu. Měřený signál má opět kmitočet 500 Hz, jehož perioda je 2 ms. Vzorkovací frekvence je však nastavena na 50 kSPS. Blok 400 vzorků, vzdálených nyní od sebe 20 µs je odebírán celkem 8 ms (20 µs * 400). Doba sledování měřeného signálu je tedy 8 ms, doba přenosu dat do počítače je stále stejná, tedy 83 ms. Celý případ je znázorněn na obr. 32. Z obou příkladů je vidět, že lépe je mít zobrazeno více period měřeného signálu, tak máme totiž větší šanci, že nám z měřeného signálu neuniknou nějaké detaily.



Obr. 32 Měření sinusového signálu 500 Hz pomocí vzorkovací rychlosti 50 kSPS

Závěr

Z textu této práce plyne, že skutečně bylo vytvořeno zařízení, které dokáže na obrazovce počítače vykreslovat průběh měřeného napětí. Zařízení je schopno zobrazovat průběh měřeného napětí o maximální amplitudě -2,5 V až +2,5 V, vyšší vstupní amplituda bude vstupním omezovačem oříznuta. Maximální frekvence vzorkování je 100 kHz. Z tohoto plyne, že pro např. kmitočet měřeného signálu 250 Hz je odebráno 400 vzorků na periodu, pro kmitočet 1 kHz máme k dispozici 100 vzorků na periodu, pro 10 kHz už jen 10 vzorků a pro 50 kHz už pouze 2 vzorky na periodu. Toto je nutné brát při měření v úvahu, abychom aliasing nepovažovali za skutečný průběh. Zařízení je vhodné pouze pro měření nízkofrekvenčních signálů.

Zařízení má navíc implementovanou funkci s jejíž pomocí můžeme snadno změnit vzorkovací rychlost, stejně jak je to možné u digitálních osciloskopů. Tato změna vzorkovací frekvence je prováděna softwarově. Zařízení totiž ani nemá, kromě dvou resetovacích tlačítek, žádné manuální ovládací prvky.

Možné rozšíření funkce stávajícího zařízení by mohlo spočívat v zařazení přesného časovacího obvodu, který by vhodně řídil vzorkování tak, aby v jedné periodě měřeného napětí byl odebrán vzorek v čase t, v další, druhé periodě by pak byl odebrán vzorek v čase t + x, ve třetí periodě pak v čase t + 2 * x, až v n-té periodě v čase t + (n - 1) * x. Čas t by se odměřoval od průchodu signálu nulovou hodnotu, například od začátku periody měřeného signálu. Časový posun x by musel být dostatečně malý a velmi přesně odměřen, po průchodu jedné periody měřeného signálu by musel být inkrementován. Tak bychom původní vzorkovací kmitočet 100 kHz mohli značně znásobit. Takto odebrané vzorky by pak byly vhodně sestaveny do výsledného tvaru měřeného signálu. Takový princip nazýváme vzorkování v ekvidistantních časových intervalech nebo zkráceně ekvivalentní vzorkování.

Tohoto zařízení by mohly například využít některé školy při výuce elektrotechniky, např. pro zobrazení průběhu napětí na diodách usměrňovače a dalších průběhů. Domácí kutilové by mohli toto zařízení např. při oživování všelijakých obvodů, kde napětí bude v rozmezí měřitelnosti zařízením. Pro vyšší napětí však stačí použít kompenzovaný dělič (1:10) a v ovládacím programu zaškrtnout políčko „1:10“.

Seznam zkratek

A/D	Analogově-digitální	(Analog to Digital)
ADC	Analogově digitální převodník	(Analog to Digital Converter)
ALU	Aritmeticko-logická jednotka	(Aritmetic-Logic Unit)
Bd	Přenosová rychlost bit za sekundu	(Baud – b/s, bit per second)
BNC	Vysokofrekvenční konektor	(Bayonet Nut Coupler)
BW	Šířka pásma	(BandWidth)
CMOS	Technologie výroby polovodičů	(Complementary Metal Oxid Semiconductor)
CPU	Mikroprocesor	(Central Processing Unit)
D/A	Digitálně-analogový	(Digital to Analog)
DAC	Digitálně-analogový převodník	(Digital to Analog Converter)
DC/DC	Měnič stejnosměrného napětí	(Direct Coupled/Direct Coupled)
DIP	Pouzdro integrovaného obvodu	(Dual Inline Package)
DNL	Diferenciální nelinearita u ADC	(Differencial Non Linearity)
DR	Dynamický rozsah	(Dynamic Range)
EEPROM	Elektricky smazatelná paměť	(Electrically Eraseable Programable Read Only Memory)
ENOB	Efektivní rozlišení u ADC	(Efficient Numer Of Bits)
EOC	Dokončení převodu napětí u ADC	(End Of the Conversion)
GND	Zem	(Ground)
HW	Fyzická část zařízení	(Hardware)
IN	Vstup	(Input)
INL	Integrální nelinearita u ADC	(Integral Non Linearity)
kSPS	Tisíc vzorků za sekundu	(Thousand Samples Per Second)
LED	Svítivé diody	(Light Emitting Diode)
LFCSP	SMD pouzdro integr. obvodu	(Lead Frame Chip Scale Package)
LQFP	SMD pouzdro integr. obvodu	(Lead Quad Flatpack Package)
LSB	Nejméně významný bit	(the Least Significant Bit)
MOSFET	tranzistor řízený polem	(Metal Oxid Semiconductor Field Effect Tran.)
MIPS	Milion instrukcí za sekundu	(Million Instruction Per Second)
MSB	Nejvíce významný bit	(the Most Significant Bit)
MSPS	Milion vzorků za sekundu	(Million Samples Per Second)
NF	Nízkofrekvenční	(NF, LF Low Frequency)

NRZ	Metoda kódování signálů,	(Non Return to Zero)
OUT	Výstup	(Output)
PC	Osobní počítač	(Personal Computer)
PulSAR	Technologie aproximačního ADC	(Pul Successive Aproximation Register)
PWM	Pulsně-šířková modulace	(Pulse Width Moduation)
OFP	SMD pouzdro integr. obvodu	(Quad Flat Package)
RAM	Paměť s náhodným přístupem	(Random Access Memory)
RISC	Redukovaný instrukční soubor	(Reduced Instruction Set Controler)
RXD	Příjem dat	(Receive Data)
S/H	Vzorkovací obvod „vzorkuj a drž“	(Sample and Hold)
S/N	Signál vůči šumu	(Signal to Noise)
S/(N+D)	Signál vůči šumu a zkreslení	(Signal/(Noise + Distortion)
SAR	Registr postupné aproximace	(Successive Aproximation Register)
SMD	Techn. povrchové montáže souč.	(Surface Mount Device)
SNR	Odstup signálu a šumu	(Signal to Noise Rejection)
SP	Ukazatel zásobníku	(Stack Pointer)
SPI	Sériové rozhraní mikrokontroléru	(Serial Pheripetial Interface)
SPS	Počet odebraných vzorků za sek.	(Samples Per Second)
SRAM	Statická paměť s náhodným příst.	(Static Random Access Memory)
SW	Program	(Software)
THD	Celkové zkreslení vyššími harm.	(Total Harmonic Distortion)
TQFP	SMD pouzdro integrovaného obv.	(Thin Quad Flat Package)
TTL	Technologie výroby polovodičů	(Transistor Transistor Logic)
TWI	Dvouvodičové rozhraní	(Two Wires Interface)
TXD	Vysílání dat	(Transmit Data)
UART	Sériová komunikační jednotka	(Universal Asynchronous Receiver and Transmiter)
USART	Sériová komunikační jednotka	(Universal Synchronous and Asynchronous Receiver and Transmiter)
USB	Univerzální sériové rozhraní PC	(Universal Serial Bus)
VCC	Napájecí napětí	(Voltage colector – colector)
VCP	Ovládací grafický program	(Virtual Control Programm)

Seznam obrázků

Seznam použité literatury

[1] Haasz, V., Sedláček, M.: Elektrická měření, Praha: ČVUT, 2003, 337 s., ISBN: 80-01-02731-7, 2. vydání.

[2] Janeček, J.: Projektování mikropočítačových systémů, Praha: ČVUT, 1999, .

[3] Matoušek, D.: Práce s mikrokontroléry ATMEL AVR – ATmega16, Praha: BEN, 2006, 320s., ISBN 80-7300-174-8, 1. vydání.

[4] Matoušek, D.: Udělejte si z PC v Delphi 1, Praha: BEN, 2003, 272s., ISBN 80-7300-111-X, 1.vydání.

[5] Matoušek, D.: USB prakticky 1 – s obvody FTDI, Praha: BEN, 2003, 272s., ISBN 80-7300-103-9, 1. vydání.

[6] Matoušek, D.: Visual C++: vývojové prostředí, Praha: BEN, 2003, 306s., ISBN 80-7300-130-6, 1. vydání.

[7] Skalický, P.: Číslicové systémy v radiotechnice, Praha: ČVUT, 2004, 201s., ISBN 80-01-02854-2, 1. vydání.

[8] Vedral, J., Fischer, J.: Elektronické obvody pro měřící techniku, Praha: ČVUT, 2004, 340 s., ISBN: 80-01-02966-2, 2. vydání.

[9] Uhlíř, J., Sovka, P.: Číslicové zpracování signálů, Praha: ČVUT, 2002, 327s., ISBN 80-01-02613-2, 2. vydání.

[10] Zaratian, B.: Visual C++ 6.0, příručka programátora, Brno: Computer press, 2004, 615s., ISBN 80-7226-151-7, 2. vydání.

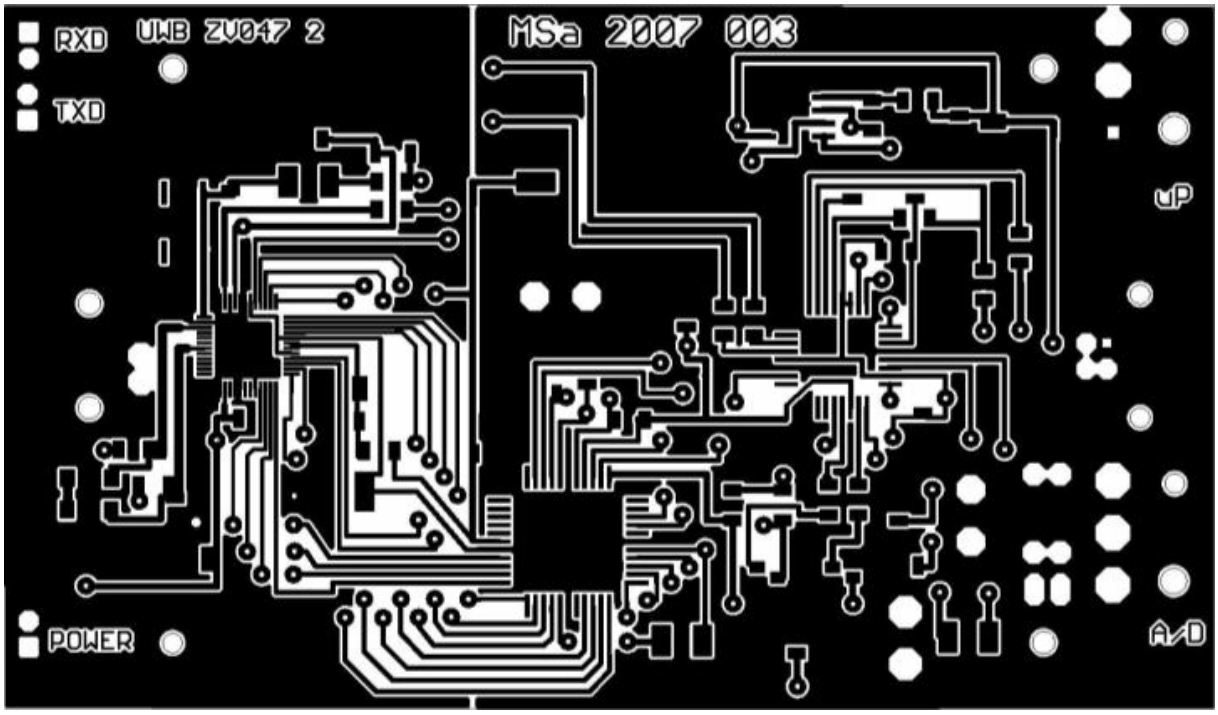
[11] ATMEL corp., [online]. c2007, aktualizováno 1.5.2007. Dostupné z: <http://www.atmel.com/dyn/resources/prod_documents/doc2502.pdf>.

[12] ANALOG DEVICES inc., [online]. c2007, aktualizováno 1.5.2007. Dostupné z: < http://www.analog.com/UploadedFiles/Data_Sheets/AD7675.pdf>.

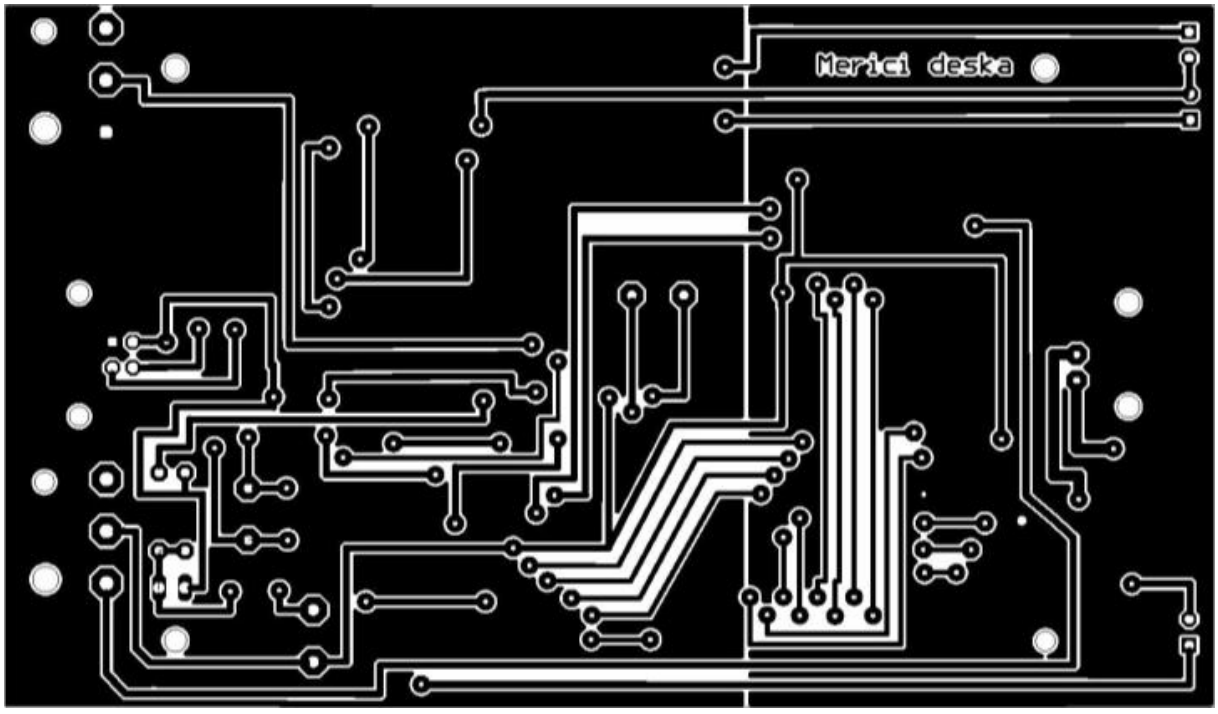
[13] FUTURE TECHNOLOGY DEVICES intl., [online]. c2007, aktualizováno 1.5.2007. Dostupné z: <http://www.ftdichip.com/Documents/DataSheets/DS_FT232BM.pdf>.

Seznam příloh

Kliše desky plošných spojů



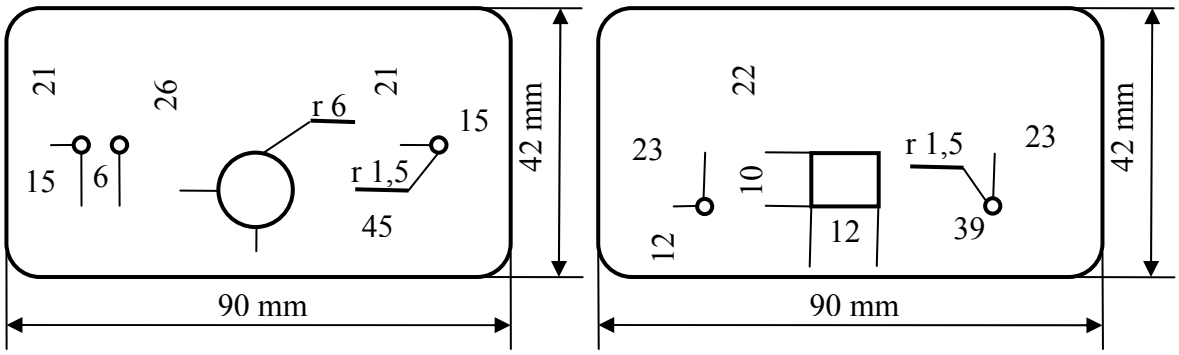
Strana součástek a spojů – TOP (300 dpi)



Strana spojů – BOTTOM (300 dpi)

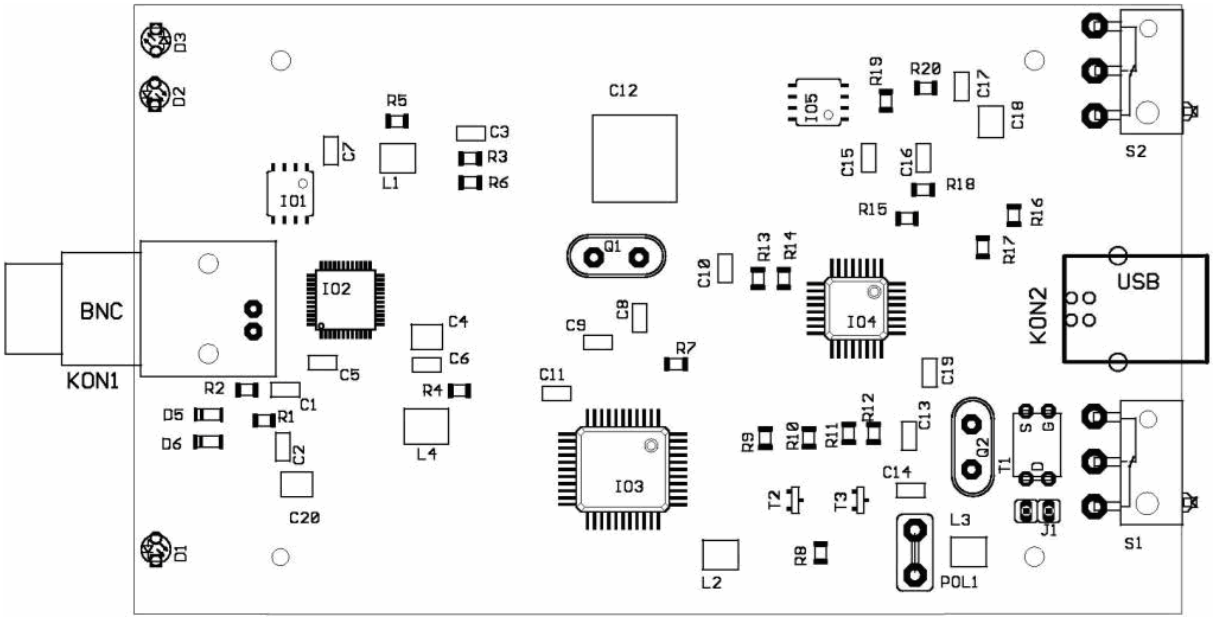
Pozn. Otvory s čtvercovým tvarem spojují zem vrstvy TOP se zemí vrstvy BOTTOM, musí tedy být zapájeny z obou stran. Takto je to řešeno jen u konektoru USB a resetovacího tlačítka „uP“. Vývod USB konektoru nelze zapájet z obou stran, což nevadí, ale proto je nutné z obou stran připájet vývod zmíněného tlačítka „uP“.

Otvory pro konektory v panelech krabičky



Přední panel **Zadní panel**

Rozmístění součástek na desce

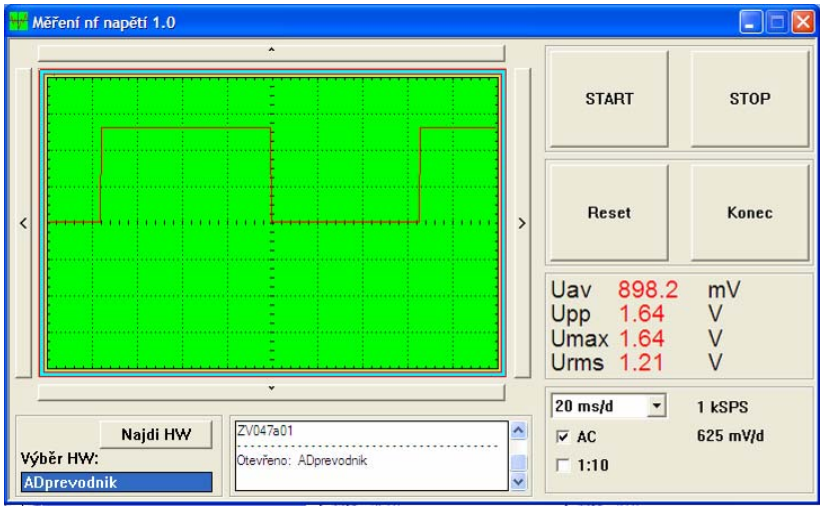


Rozvržení součástek na desce (118 mm x 67 mm)

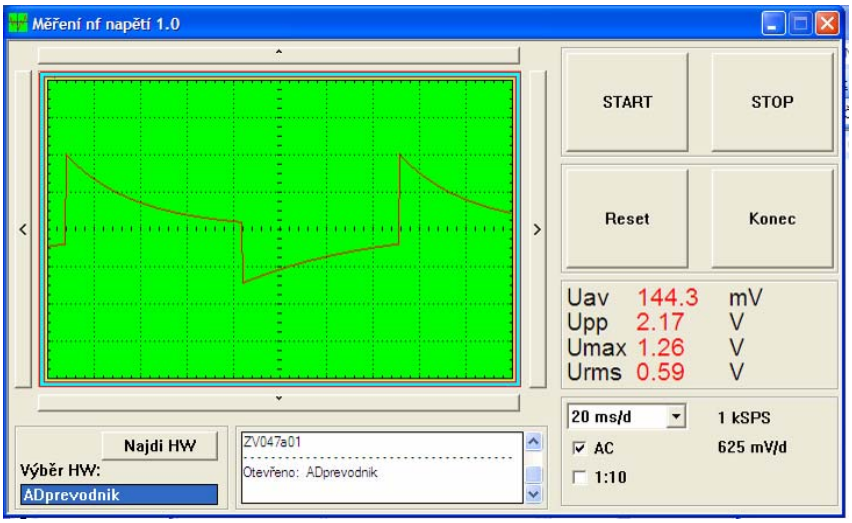
Seznam součástek pro desku UWB_ZV047_2

Rezistory SMD, vrstvové:		
R1, R2	150R	vel. 1206
R3, R4	100R	vel. 1206
R5, R7	1k	vel. 1206
R6	50R	vel. 1206
R8, R15	470R	vel. 1206
R9, R10, R11, R12, R20	10k	vel. 1206
R13,R14	220R	vel. 1206
R16, R17	27R	vel. 1206
R18	1k5	vel. 1206
R19	2k2	vel. 1206
Kondenzátory SMD:		
C1, C2	2n7	vel. 1206, keram.
C3, C19	10n	vel. 1206, keram.
C4, C18, C20	10M/16V	vel. B, tantalový
C5, C6, C10, C11, C16, C17	100n	vel. 1206, keram.
C7	1n	vel. 1206, keram.
C8, C9, C13, C14	27p	vel. 1206, keram.
C12	470M/16V	vel. F, hliníkový
C15	33n	vel. 1206, keram.
Tlumivky SMD:		
L1, L2, L3	33M	JCI
L4	220M	TDR
Krystaly:		
Q1	16 MHz	
Q2	6 MHz	
Integrované obvody, SMD:		
IO1	TL431	SOP 8
IO2	AD7675	LQFP 48
IO3	ATmega16	TQFP 44
IO4	FT232BM	LQFP 32
IO5	93LC46	SOP 8
Tranzistory:		
T1	IRFD9120	
T2, T3	BC848C	SOT23 (SMD)
Diody:		
D1, D2	LED, zelená	d = 3 mm
D3	LED, červená	d = 3 mm
D4, D5	MMZ2,7, zenerova	SOD80 (SMD)
Ost;atní:		
POL1	Polyswitch, 500 mA	
KON1	BNC do PS 90°	
KON2	USB B do PS 90°	
S1, S2	Tlačítko do PS 90°	
J1	Jumper	
Krabička	KG B10	95x135x45 mm

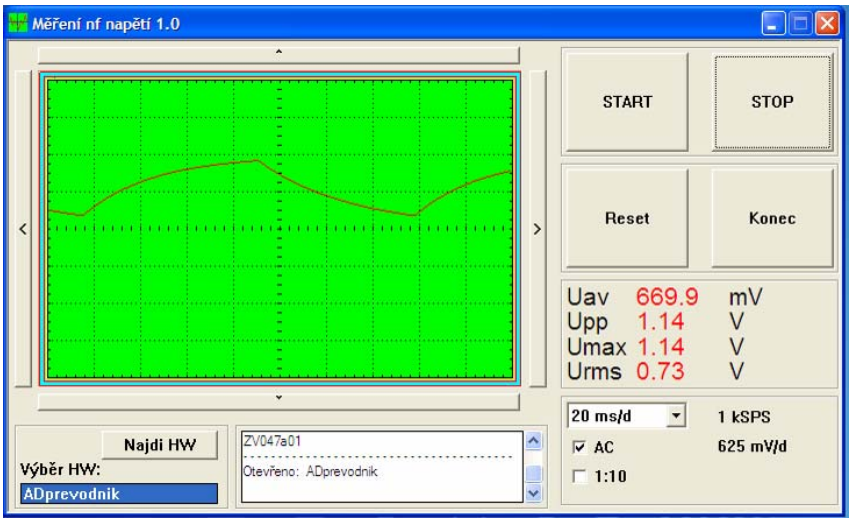
Výsledky měření



Vstupní obdélníkový signál 7 Hz

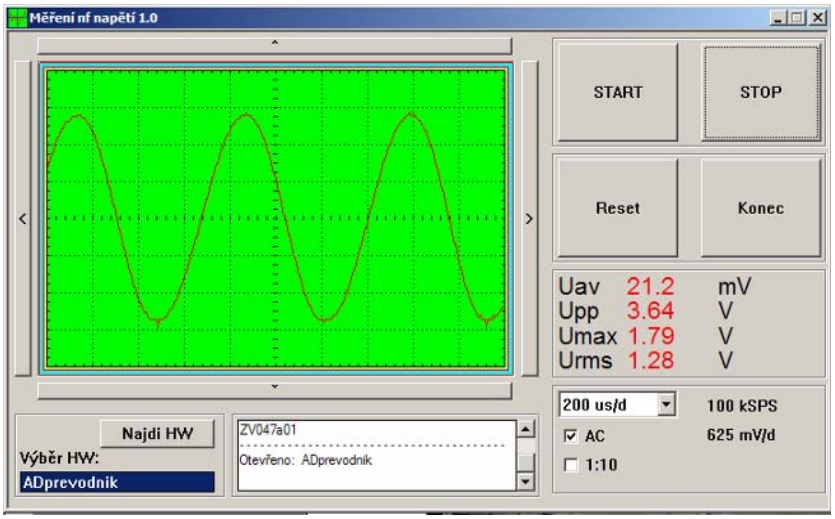


Výstupní signál derivačního členu

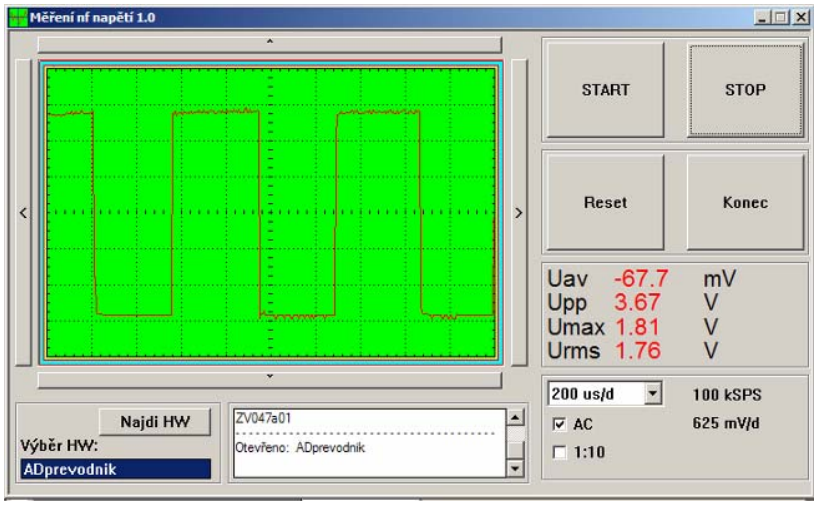


Výstupní signál integračního členu

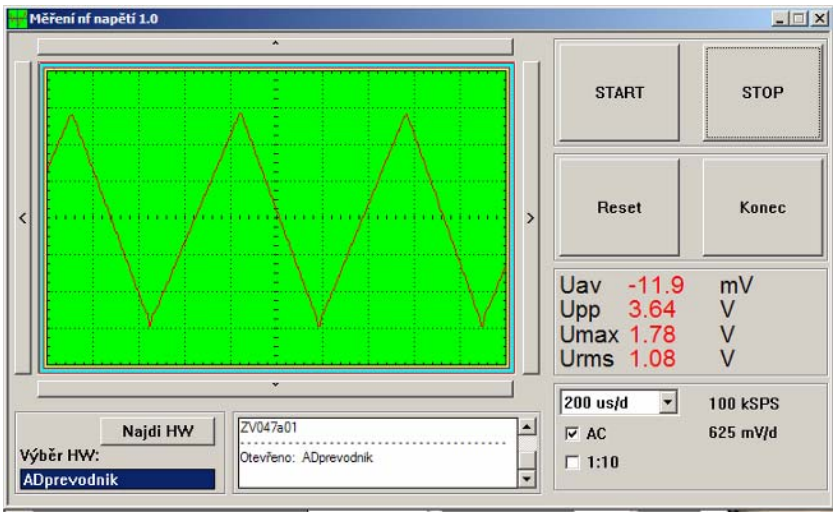
Výsledky měření



Sinusový signál 1 kHz, vzorkován 100 kSPS

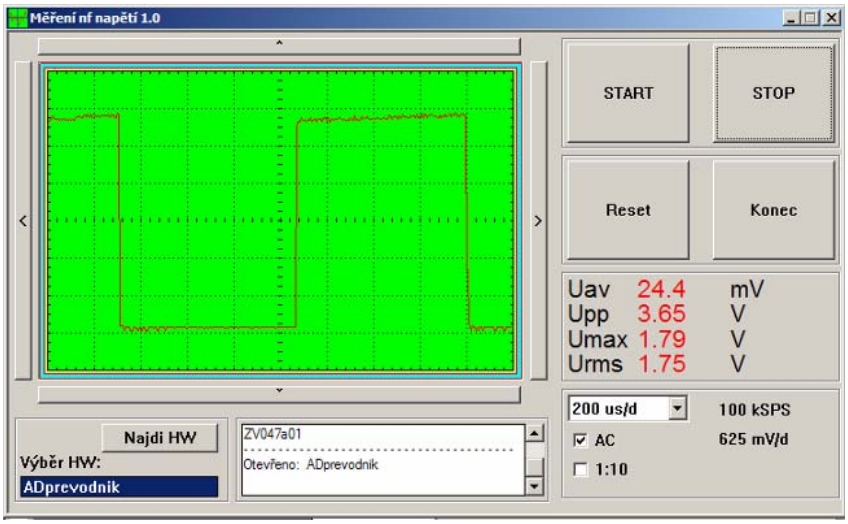


Obdélníkový signál 1 kHz, vzorkován 100 kSPS

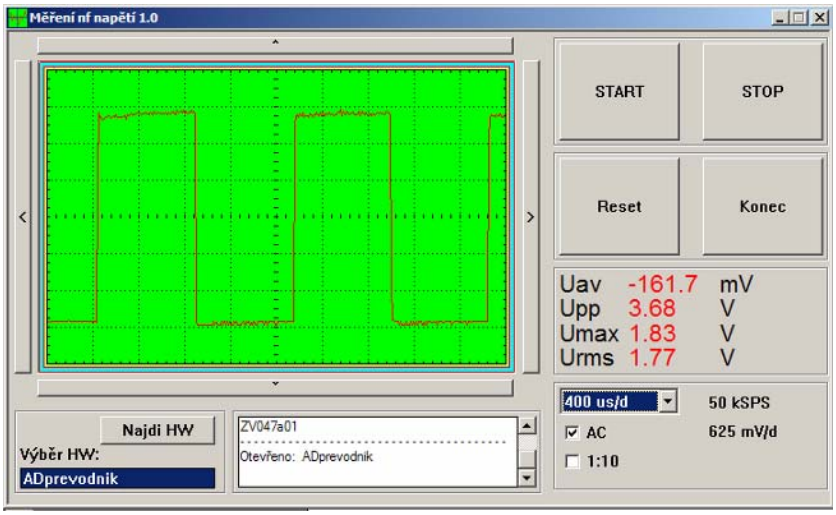


Pilovitý signál 1 kHz, vzorkován 100 kSPS

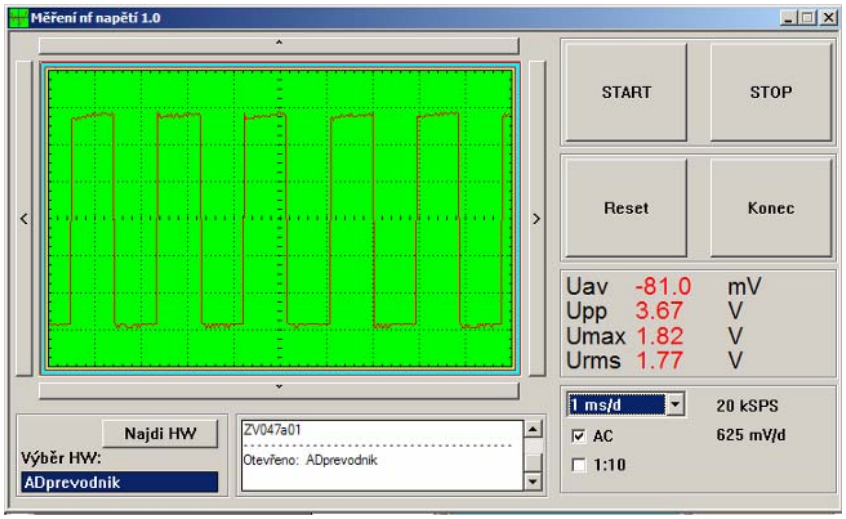
Výsledky měření



Obdélníkový signál 550 Hz, vzorkován 100 kSPS

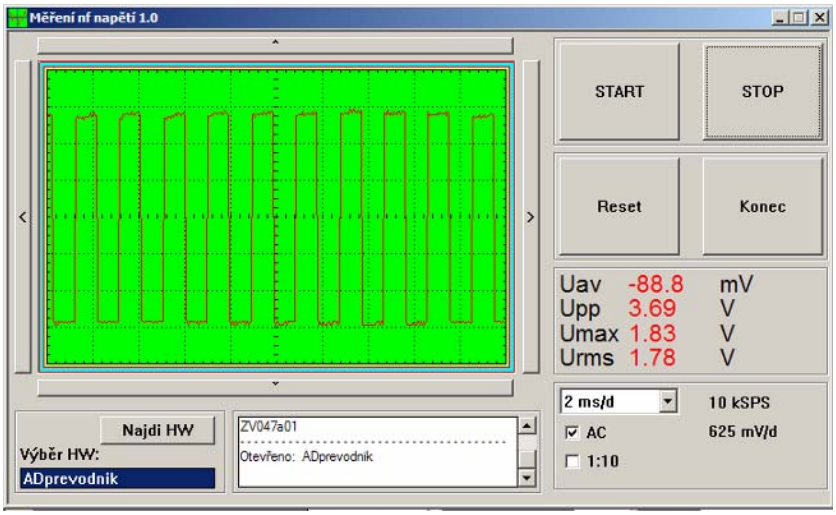


Obdélníkový signál 550 Hz, vzorkován 50 kSPS

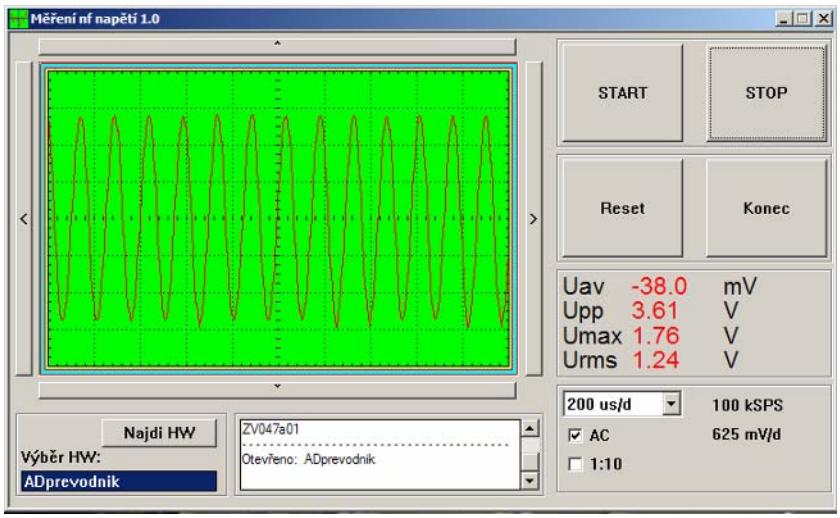


Obdélníkový signál 550 Hz, vzorkován 20 kSPS

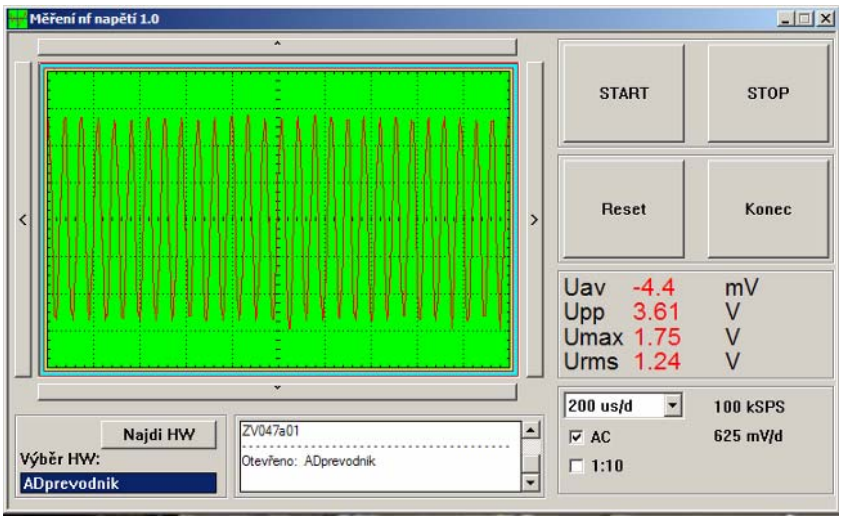
Výsledky měření



Obdélníkový signál 550 Hz, vzorkován 10 kSPS

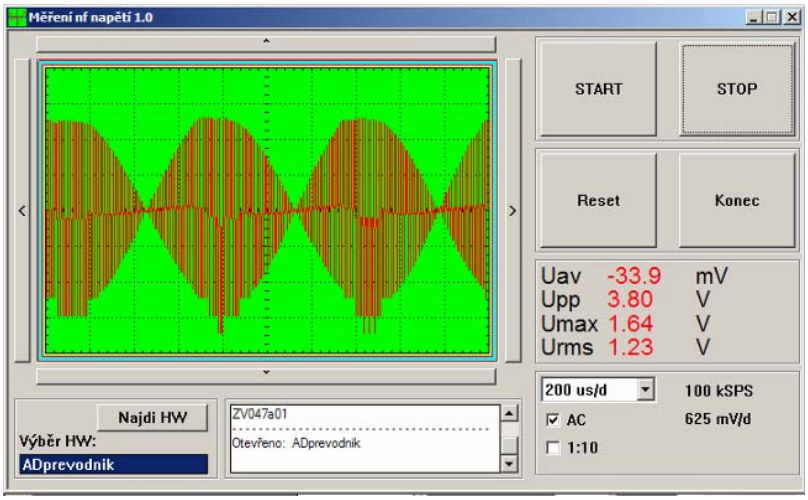


Sinusový signál cca 6 kHz, vzorkován 100 kSPS

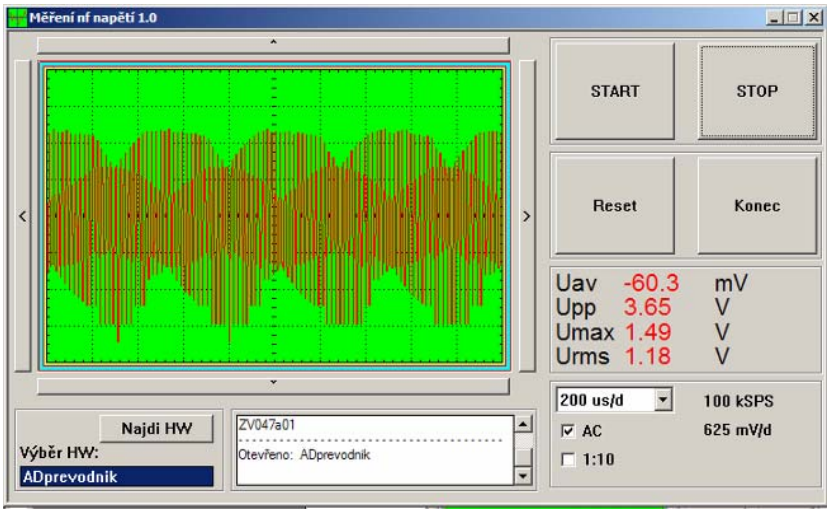


Sinusový signál cca 15 kHz, vzorkován 100 kSPS

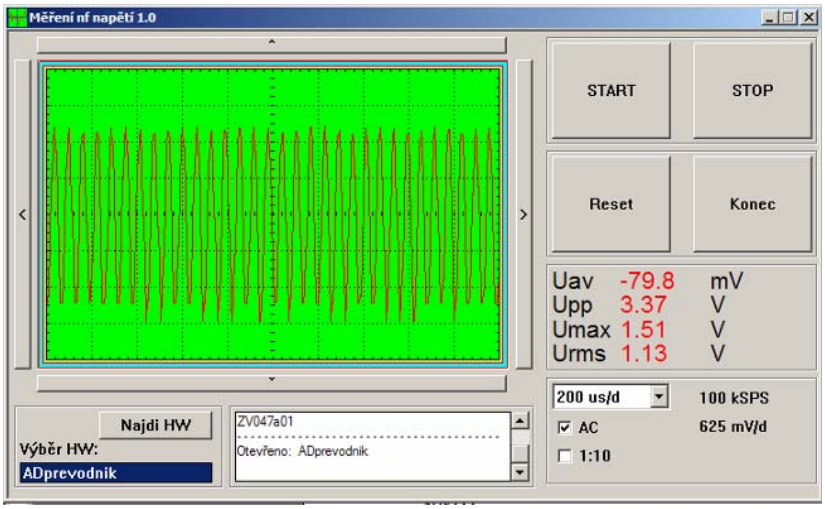
Výsledky měření – ALIASING



Sinusový signál 60 kHz, vzorkován 100 kSPS

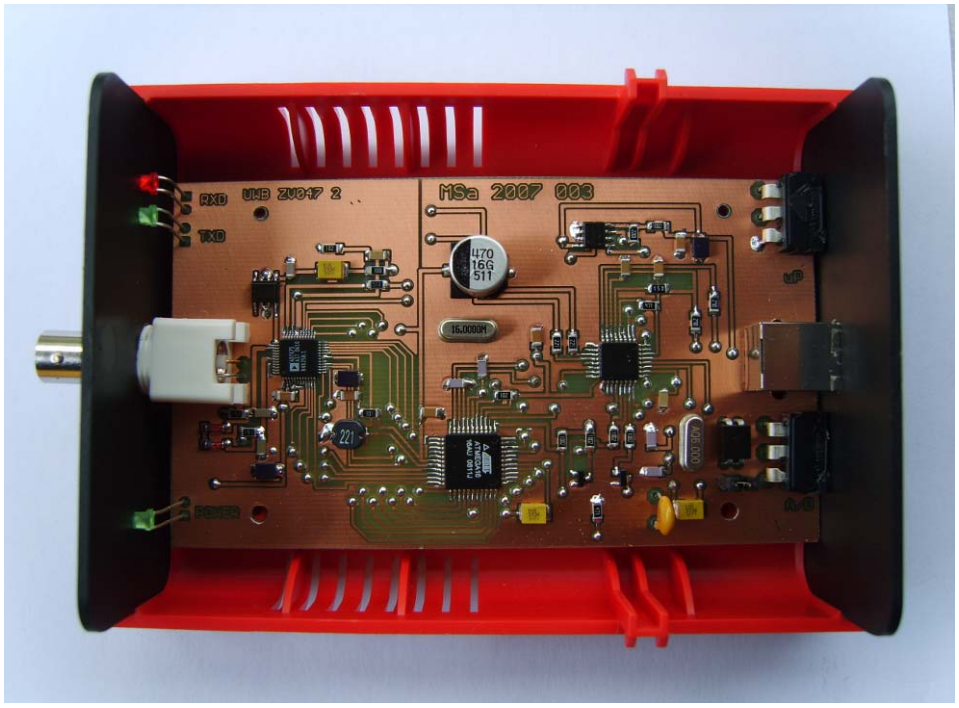


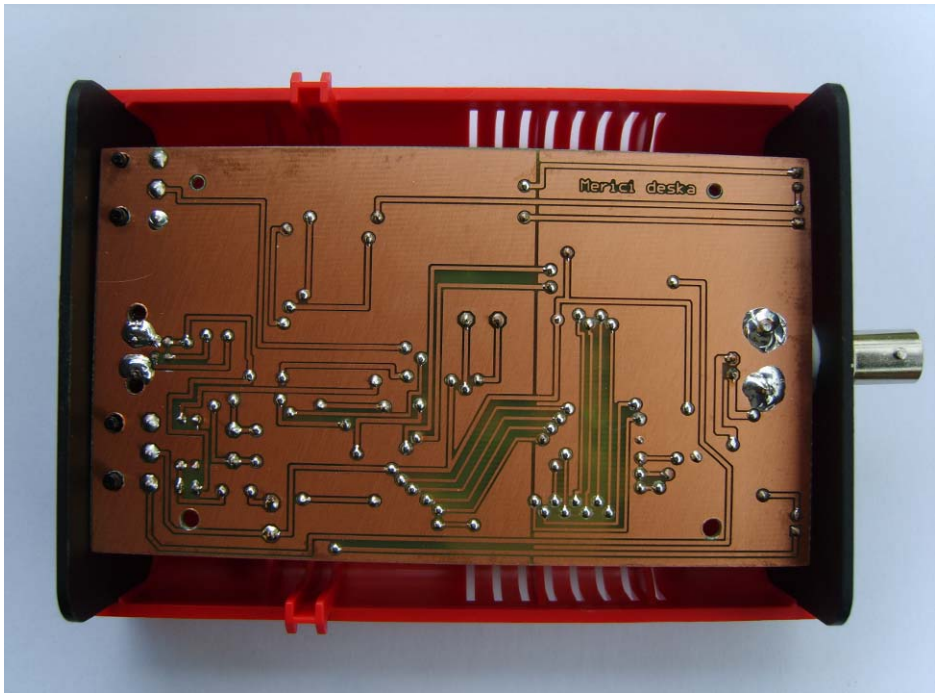
Sinusový signál 80 kHz, vzorkován 100 kSPS



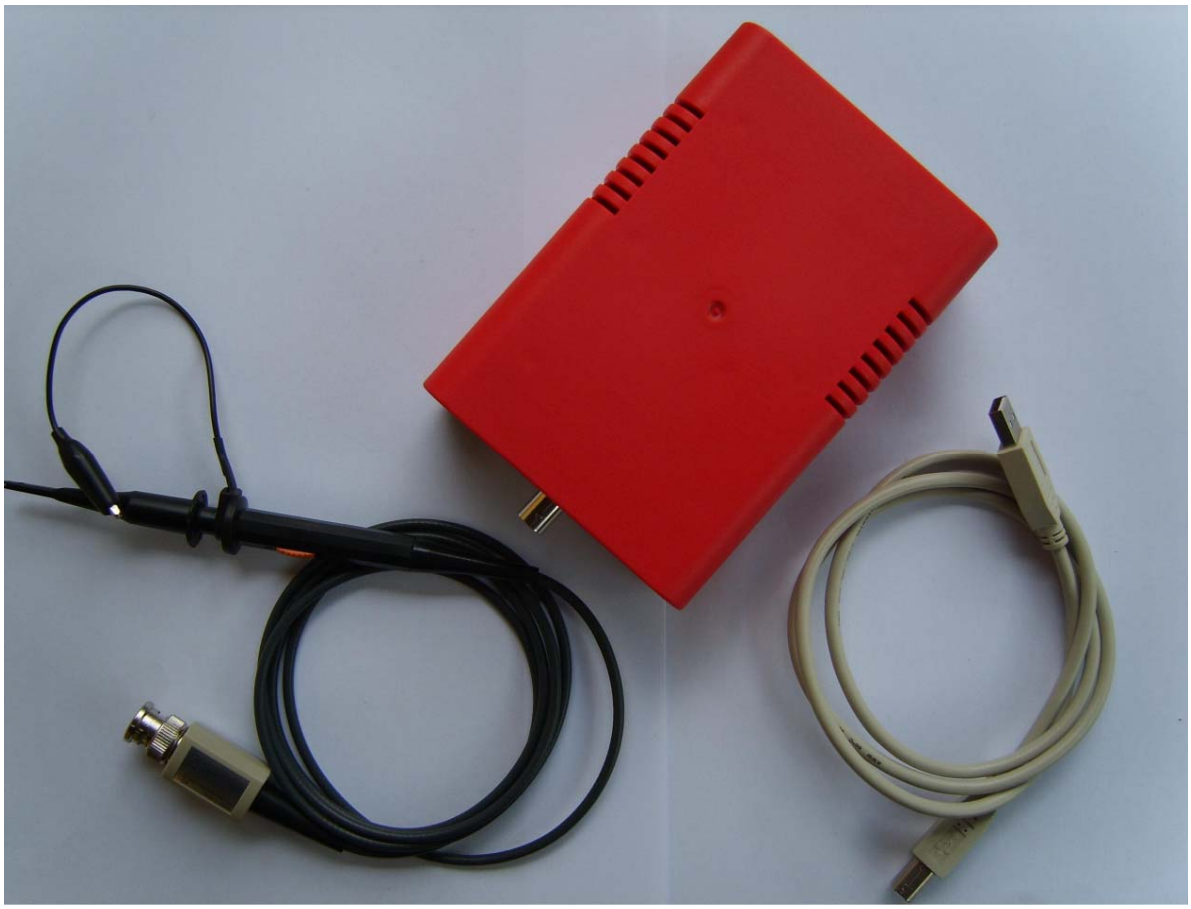
Sinusový signál 100 kHz, vzorkován 100 kSPS

Vyobrazení zařízení





Vyobrazení zařízení



Výpis assembleru pro mikrokontrolér

```
        .NOLIST
        .INCLUDE "m16def.inc"
        .LIST
        .EQU BAUD=8
        .DSEG
        .ORG $60

DATA:
        .BYTE(802)
        .CSEG

; Nastavení SP na konec RAM
RESET:
        LDI R16,LOW(RAMEND)
        OUT SPL,R16
        LDI R16,HIGH(RAMEND)
        OUT SPH,R16

; Nastavení sériového přenosu do PC
        LDI R16,HIGH(BAUD)
        OUT UBRRH,R16

        LDI R16,LOW(BAUD)
        OUT UBRRL,R16

        LDI R16,(1<<RXEN)|(1<<TXEN)
        OUT UCSRB,R16

        LDI R16,(1<<URSEL)|(1<<UCSZ0)|(1<<UCSZ1)
        OUT UCSRC,R16

; Nastavení portů
        CLR R16
        OUT DDRC,R16
        OUT DDRA,R16
        SER R16
        OUT DDRB,R16
        CBI DDRB, 2
        CBI DDRB, 0
        SBI DDRD, 1

 ; Nezapínej AD převod
        SBI PORTB, 1
        LDI R19, 0
        LDI R21, 255
        LDI R20, 128

; Čekání na povel z počítače
RADC:
        IN R16,UCSRA
        ANDI R16,0b10000000
        CPSE R16, R20
        RJMP RADC

AD:
; Přečti hodnotu rychlosti vzorkování
        IN R25, UDR
```

```
; Nastavení ukazatele Z na začátek
        LDI ZL, LOW(DATA)
        LDI ZH, HIGH(DATA)

; Počítací registry 401 vzorků
        LDI R17, 252
        LDI R18, 150

; Start převodu AD
        CBI PORTB, 1

; Obnovení registru s požadovanou rychlostí vzorkování
CTI:
        MOV R29,R25

; Načtení datových bajtů z ADC
        IN R23, PINA
        IN R22, PINC

; Eliminace horních datových bajtů s hodnotou 128
        CPSE R23, R20
        RJMP DALE
        INC R23

; Eliminace dolních datových bajtů s hodnotou 128
DALE:
        CPSE R22, R20
        RJMP ULOZ
        INC R22

; Ukládaní vzorků do paměti s časováním odběru
ULOZ:
        ST Z+, R23
       ST Z+, R22
        RCALL CEKEJ
        DEC R17
        BREQ DALSI_DEC
        RJMP CTI

; Počítací rutina pro uložení potřebného počtu vzorků
DALSI_DEC:
        LDI R17, 1
        DEC R18
        CPSE R18, R19
        RJMP CTI
        RJMP POSLIDATA

        ; Naplnění počítacích registrů
POSLIDATA:
        LDI R17, 252
        LDI R18, 150

        ; Nastavení ukazatele Z na začátek
        LDI ZL, LOW(DATA)
        LDI ZH, HIGH(DATA)

; Odeslání synchronizačního bajtu
SYNCHROBAJT1:
        SBIS UCSRA,UDRE
        RJMP SYNCHROBAJT1
```

```
        OUT UDR, R20

; Odeslání synchronizačního bajtu
SYNCHROBAJT2:
        SBIS UCSRA,UDRE
        RJMP SYNCHROBAJT2
        OUT UDR, R20

; Načítání bajtů z paměti mikrokontroléru
NACTI:
        LD R23, Z+
        LD R22, Z+

; Odeslání synchronizačního bajtu
SYNCHROBAJT3:
        SBIS UCSRA,UDRE
        RJMP SYNCHROBAJT3
        OUT UDR, R20

; Odeslání horního datového bajtu
HBYTE:
        SBIS UCSRA,UDRE
        RJMP HBYTE
        OUT UDR,R23

; Odeslání spodního datového bajtu
LBYTE:
        SBIS UCSRA,UDRE
        RJMP LBYTE
        OUT UDR,R22

        DEC R17
        BREQ DALSI_DEC1
        RJMP NACTI

; Počítací rutina pro odesílání potřebného počtu vzorků
DALSI_DEC1:
        LDI R17, 1
        DEC R18
        CPSE R18, R19
        RJMP NACTI
        RJMP RADC

; Čekací rutina pro časování vzorkování
CEKEJ:
        LDI R16, 26
A:
        DEC R16
        BRNE A
        DEC R29
        BRNE CEKEJ
    RET
```

ÚDAJE PRO KNIHOVNICKOU DATABÁZI

Název práce	Měření nf signálů pomocí A/D převodníku s vysokým rozlišením
Autor práce	Martin Šafařík
Obor	Elektronika
Rok obhajoby	2007
Vedoucí práce	Ing. Jan Mrkvica, Ph.D.
Anotace	Práce se obecně věnuje A/D převodníkům a popisuje stavbu zařízení pro měření nízkofrekvenčních signálů, včetně popisu řídícího softwaru
Klíčová slova	A/D převodník Mikrokontrolér Visual C++ Assembler Vzorkování Osciloskop

FORMULÁŘ PRO ZPŘÍSTUPNĚNÍ PRÁCE V ELEKTRONICKÉ FORMĚ – ČESKY

Typ dokumentu	*Bakalářská práce*		
Autor	**Martin Šafařík**		
E-mail. adresa autora	mar.saf@seznam.cz		
URN			
Název závěrečné práce	*Měření nf signálů pomocí A/D převodníku s vysokým rozlišením.*		
Stupeň studia	*Bakalářské*		
Katedra	**Katedra elektroniky, elektrotechniky a zabezpečovací techniky v dopravě**		
Vedoucí práce	**Ing. Jan Mrkvica, Ph.D., vedoucí práce**		
Klíčová slova	*A/D převodník* *Mikrokontrolér* *Visual C++* *Assembler* *Vzorkování* *Osciloskop*		
Datum obhajoby	*2007-06-25*		
Označení rozsahu zpřístupnění	*souhlasím se zveřejněním celé práce*	Datum: Podpis autora:	
Abstrakt	*Práce se obecně věnuje A/D převodníkům a popisuje stavbu zařízení pro měření nízkofrekvenčních signálů, včetně popisu řídícího softwaru*		
Název souboru	*Safarik_bc_07.pdf*	Velikost souboru	5 MB

FORMULÁŘ PRO ZPŘÍSTUPNĚNÍ PRÁCE V ELEKTRONICKÉ FORMĚ – ANGLICKY

Type of Document	*Bachelor work*		
Author	**Martin Šafařík**		
Author's E-mail Address	mar.saf@seznam.cz		
URN			
Title	*Low frequency signals measurment using high resolution AD converter*		
Degree	*Bachelor*		
Department	**Department of electrical and electronical engineering and signalling in transport**		
Advisory Committee	**Ing. Jan Mrkvica, Ph.D., supervisor**		
Keywords	*A/D converter* *Mikrocontroller* *Visual C++* *Assembler* *Sampling* *Oscilloscope*		
Date of Defense	*2007-06-25*		
Availability	*Unrestricted*		
Abstract	*A work is about analog to digital conversion and describe how to make the low frequency signals measurmnet device, including control programme describe*		
Filename	*Safarik_bc_07.pdf*	Size	5 MB