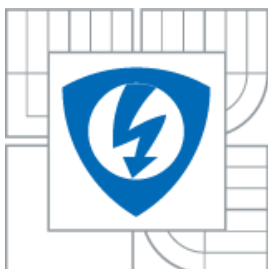




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

IMPULZOVÉ MODULACE

PULSE MODULATIONS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

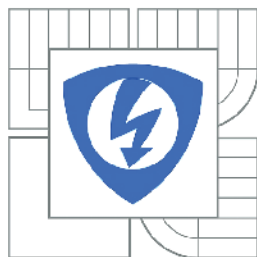
AUTOR PRÁCE
AUTHOR

Bc. MARTIN ŠIŠKA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. RADIM ČÍŽ, Ph.D.

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Martin Šiška

ID: 115285

Ročník: 2

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Impulzové modulace

POKYNY PRO VYPRACOVÁNÍ:

V prostředí Matlab-Simulink vytvořte demonstrační modely impulzových modulací PAM, PPM, PWM, PCM, DPCM, DM, ADM, SDM. Navrhněte koncepce a podrobná schémata zapojení sady laboratorních přípravků (modemů), které umožní porovnání vybraných druhů modulací a jejich parametrů. Zhotovte funkční vzorky těchto přípravků a připravte kompletní dokumentaci pro jejich kusovou výrobu.

DOPORUČENÁ LITERATURA:

- [1] SKLAR, B. Digital Communications. 2nd. ed. Upper Saddle River : Prentice Hall, 2001. 1080 p. ISBN 0-13-084788-7.
- [2] DOBEŠ, J., ŽALUD, V. Moderní radiotechnika. Praha : BEN, 2006. 768 s. ISBN 80-7300-132-2.
- [3] MATOUŠEK, D. Práce s mikrokontroléry Atmel AVR . Praha : BEN, 2006. 376 s. ISBN 80-7300-209-4.
- [4] ŠEBESTA, V. Teorie sdělování. Brno : VUTUM, 1998. 92 s. ISBN 80-214-1247-X.

Termín zadání: 11.2.2013

Termín odevzdání: 29.5.2013

Vedoucí práce: Ing. Radim Číž, Ph.D.

Konzultanti diplomové práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

Abstrakt

Tato diplomová práce se zabývá rozбором problematiky impulzových modulací jejich simulacemi a jejich programováním.

První kapitola je věnována impulzovým modulacím jako celku. Je zde vysvětlen pojem modulace, uveden rozdíl mezi analogovou a digitální modulací a také je zde uvedeno základní dělení impulzových modulací.

Ve druhé kapitole se práce zaměřuje na problematiku nekvantovaných impulzových modulací. Každá modulace, popisovaná v této je slovně popsána, přičemž je tento slovní popis pro lepší představivost doplněn graficky.

Ve třetí kapitole, která je obdobou kapitoly druhé, se práce zaměřuje na kvantované impulzní modulace. Opět je u každé z těchto modulací vysvětlen její základní princip. Dále je také v této kapitole nastíněn návrh modelů v prostředí Matlab-Simulink.

Ve čtvrté kapitole se práce zabývá porovnáním časových průběhů signálů získaných simulacemi.

V páté kapitole je popsán návrh programů realizující zvolené modulace/demodulace a také návrh obslužné aplikace. Stejně tak jsou v této kapitole zachyceny časové průběhy získané měřením naprogramovaných modulací a demodulací.

Šestá kapitola se věnuje porovnání simulací impulzových modulací a měření jejich naprogramovaných verzí.

V sedmé kapitole se práce věnuje návrhu koncepcí pro výrobu laboratorních přípravků. Je zde také provedena diskuze vhodnosti jednotlivých koncepcí.

Osmá kapitola je věnována návrhu zapojení jednotlivých přípravků, návrhu desek plošných spojů a návrhu krabice.

V poslední, deváté kapitole, jsou uvedeny vzory výpočtů některých parametrů a tabulky vypočítaných hodnot.

Klíčová slova

Impulzové modulace
Pulzně amplitudová modulace
Pulzně šířková modulace
Pulzně polohová modulace
Pulzně kódová modulace
Diferenční pulzně kódová modulace
Delta modulace
Sigma delta modulace
Adaptabilní delta modulace
ATmega16
A/D převodník
D/A převodník
Rozhraní SPI

Abstract

This master's thesis deals with the analysis of the problems of pulse modulation, their simulations and their programming.

The first chapter is devoted to pulses modulations as a whole. It explains the concept of modulation, the distinction between analog and digital modulation, and there is the basic classification of pulse modulation stated.

The second chapter focuses on the issue of non-quantized pulse modulation. Each modulation is verbally described, for better imagination is the verbal description added graphically.

In the third chapter, which is similar to the second chapter, the work focuses on quantized pulse modulation. Again, in each of these modulations explained its basic principle. It is also in this chapter outlines the design models in Matlab-Simulink.

In the fourth chapter, the thesis deals with the comparison of waveforms obtained by simulation.

The fifth chapter describes the design of programs implementing the selected modulation / demodulation and design service application. In this chapter are also included captured waveforms obtained by measuring the programmed modulation and demodulation.

The sixth chapter deals with the comparison results of simulation and measurement of pulse modulation of programmed versions.

In the seventh chapter deals this thesis with design concepts for the manufacture of laboratory products. There is also a discussion of the suitability the concepts.

The eighth chapter is devoted to the design of involvement of individual products, PCB design and design of the box.

In the last, ninth chapter, are stated the calculations of some parameters and tables of calculated values.

Keywords

Pulse modulations
Pulse amplitude modulation
Pulse width modulation
Pulse position modulation
Pulse code modulation
Differential pulse code modulation
Delta modulation
Sigma delta modulation
Adaptive delta modulation
ATmega16
A/D converter
D/A converter
SPI interface

Citace práce

ŠIŠKA, M. Impulzové modulace. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 101 s. Vedoucí diplomové práce Ing. Radim Číž, Ph.D.

Prohlášení o původnosti

Prohlašuji, že svou diplomovou práci na téma Impulzové modulace jsem vypracoval samostatně pod vedením vedoucího diplomové práce s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny uvedeny v seznamu literatury na konci práce.

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

V Brně dne 28.5.2013

.....
(podpis autora)

Poděkování

Děkuji vedoucímu diplomové práce Ing. RADIMU CÍŽOVI, Ph.D. za velmi užitečnou metodickou pomoc a cenné rady při zpracování práce.

V Brně dne 28.5.2013

.....
(podpis autora)

Obsah

Úvod.....	7
1. Impulzové modulace	7
1.1 Pojem MODULACE	7
1.2 Rozdíl mezi analogovou modulací a impulzní modulací	7
1.3 Rozdělení impulzových modulací	8
2. Nekvantované impulzové modulace	9
2.1 Pulzně amplitudová modulace (PAM)	9
2.2 Pulzně polohová modulace (PPM)	12
2.3 Pulzně šířková modulace (PWM)	13
3. Kvantované modulace	14
3.1 Pulzní kódová modulace (PCM)	15
3.2 Diferenční pulzní kódová modulace (DPCM)	19
3.3 Delta modulace (DM).....	23
3.4 Sigma-Delta modulace Σ - Δ (SDM).....	27
3.5 Adaptabilní Delta modulace	29
4. Porovnání časových průběhů signálů získaných simulacemi	35
4.1 Porovnání časových průběhů PCM	35
4.2 Porovnání časových průběhů DPCM	37
4.3 Porovnání časových průběhů DM a ADM	39
5. Programování impulzových modulací	40
5.1 Programování modulace PCM	40
5.2 Programování Diferenční Pulzně Kódové Modulace (DPCM).....	56
5.3 Programování Delta Modulace (DM)	63
5.4 Programování Adaptabilní Delta Modulace (ADM).....	69
5.5 Programování obslužné aplikace pro jednotlivé přípravky	74
6. Porovnání simulací impulzových modulací a měření jejich naprogramovaných verzí	78
7. Návrh koncepce pro výrobu demonstračních přípravků	81
8. Návrh zapojení a výroba laboratorních přípravků	82
9. Vzor výpočtu a tabulky vypočítaných hodnot	84
Závěr.....	87
Přílohy	92
Literatura	97
Seznam obrázků	98
Seznam tabulek	100
Seznam použitých zkratk.....	101

Úvod

Prvním cílem této práce bylo podrobné seznámení se s principy impulzových kódových modulací PCM, DPCM, DM, ADM, SDM a prostudování možností realizace jednotlivých modulátorů a demodulátorů.

Dalším cílem bylo vytvoření demonstračních modelů impulzových modulací v prostředí Matlab-Simulink a provedení jejich simulací.

Dále bylo cílem navrhnout koncepce a podrobná schémata zapojení sady laboratorních přípravků, které umožní porovnání vybraných druhů modulací a jejich parametrů.

Posledním cílem práce bylo zhotovení těchto přípravků a příprava kompletní dokumentace pro jejich kusovou výrobu

1. Impulzové modulace

V této krátké kapitole bude nejdříve vysvětlen pojem modulace. Dále bude popsán rozdíl mezi analogovou a impulzní modulací. Následně bude provedeno rozdělení impulzových modulací.

1.1 Pojem MODULACE

Modulace je ovlivňování parametru nosného signálu signálem modulačním. U klasických (analogových) modulací se používá jako nosný signál vysokofrekvenční harmonická vlna, která je modulována analogovým modulačním signálem. Oba tyto průběhy jsou spojitě v čase, v amplitudě a také v kmitočtu, respektive ve fázi. [1]

1.2 Rozdíl mezi analogovou modulací a impulzní modulací

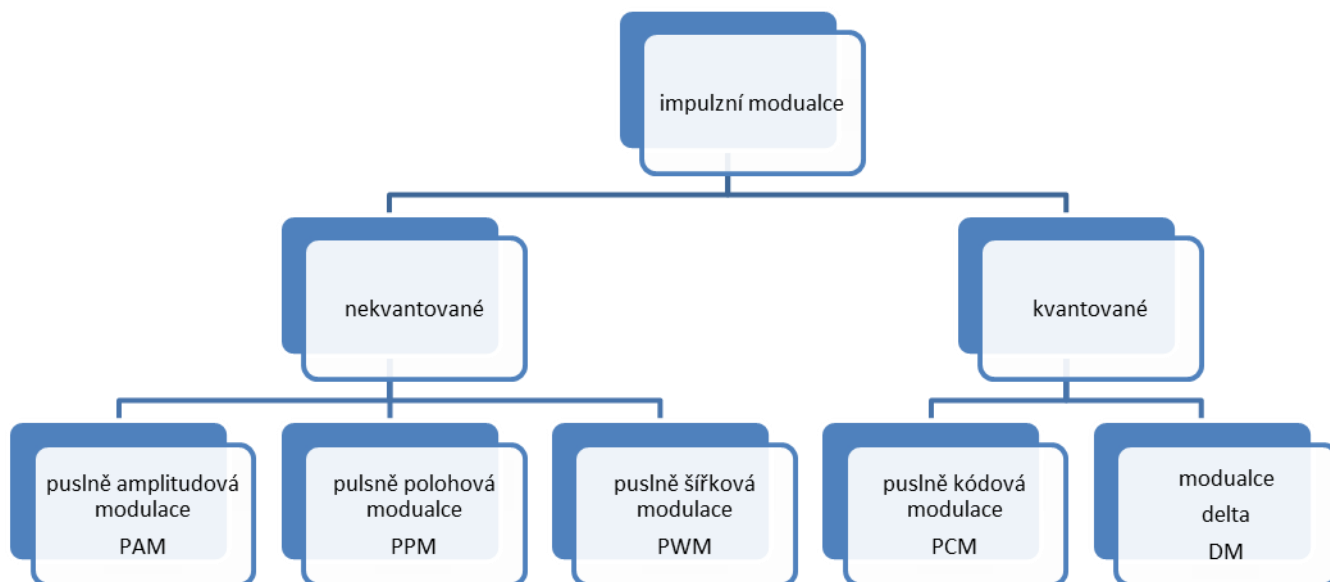
Na rozdíl od analogových modulací, používají impulzní modulace jako nosný signál periodickou posloupnost obdélníkových impulzů. Ačkoliv se jeví princip modulace s harmonickým nosným signálem a modulace impulzové velmi podobné, mají tyto modulace velmi málo společného a jejich role ve sdělovací technice je velmi odlišná.

Analogové modulace se používají ve sdělovací technice zejména pro přenos informací pomocí vysokofrekvenčního signálu, kdyžto impulzové modulace se nejčastěji využívají v D/A převodnících, nebo pro zdrojové kódování. [4]

1.3 Rozdělení impulzových modulací

Impulzové modulační lze nejjednodušším způsobem rozdělit na **nekvantované** a **kvantované**.

Výše zmíněné rozdělení je přehledně zobrazeno na obrázku č.1.



Obrázek č.1: Základní rozdělení impulzních modulací

Jak je tedy z tohoto obrázku patrné dělí se obě hlavní skupiny, tedy kvantované i nekvantované impulzní modulační dále.

U nekvantovaných modulací se dělí podle parametru nosného impulsu, který je ovlivňován. Z toho tedy vyplývá, že nekvantované modulační jsou obdobou základní analogových modulací. Mezi nekvantované metody impulzových modulací patří *Pulsně amplitudová modulační (PAM)*, *Pulsně šířková modulační (PŠM)*, *Pulsně polohová modulační (PPM)*.

Mezi kvantované metody impulzových modulací jsou *Pulsně kódová modulační* a její modifikace (*Diferenční pulsně kódová modulační...*), a *Delta modulační* se svými modifikacemi (*Sigma-delta modulační, adaptabilní delta modulační*).

Z historického hlediska byly impulzové modulační nejprve realizovány v základním kmitočtovém pásmu. U těchto modulací byl analogový modulační signál, ležící v základním pásmu, převáděn na diskrétní (impulzový) signál, který je rovněž realizován v základním pásmu. Tyto diskrétní signály se nejčastěji přenášely metalickými spoji, v pozdější době, též pomocí optických vlnodů. V případě, že mají být tyto signály přenášeny bezdrátově, je nutné je nejprve namodulovat na vysokofrekvenční nebo mikrovlnnou nosnou vlnu. Tím dostaneme diskrétní modulační s nosnými vlnami, které označujeme jako digitální modulační.[1]

2. Nekvantované impulzové modulace

V následující kapitole se tato práce věnuje skupině nekvantovaných modulací.

Principem těchto modulací je v drtivé většině pouze vzorkování. Existuje několik různých typů vzorkování, ovšem téměř všechny mají stejný základ. Tím základem je vzorkování metodou *Sample and Hold*. Tento princip je velmi jednoduchý a bude popsán v následujícím odstavci.

Vzorkování

Vzorkování je prováděno pomocí zařízení, vzorkovače, který pracuje na principu uchování aktuální hodnoty signálu po určitou dobu. Tento způsob se nazývá *Sample and Hold*. Takto se vytváří z analogového vstupního signálu vzorky, které jsou od sebe vzdáleny o dobu odpovídající vzorkovací periodě. Velikost vzorkovací periody musí být zvolena tak, aby byla zaručena dostatečná hustota vzorků, která dovoluje převod spojitého signálu na diskrétní bez ztráty informace a bez zkreslení. Tímto problémem se zabývá Shannon-Kotělnikovův teorém, který říká, že minimální vzorkovací frekvence musí být při nejmenším dvojnásobná, než jaká je maximální frekvence obsažená ve vzorkovaném signálu.

$$T_{VZ} = \frac{1}{f_{VZ}}, \quad (1)$$

kde T_{VZ} (s) je vzorkovací perioda,
 f_{VZ} (Hz) je vzorkovací frekvence.

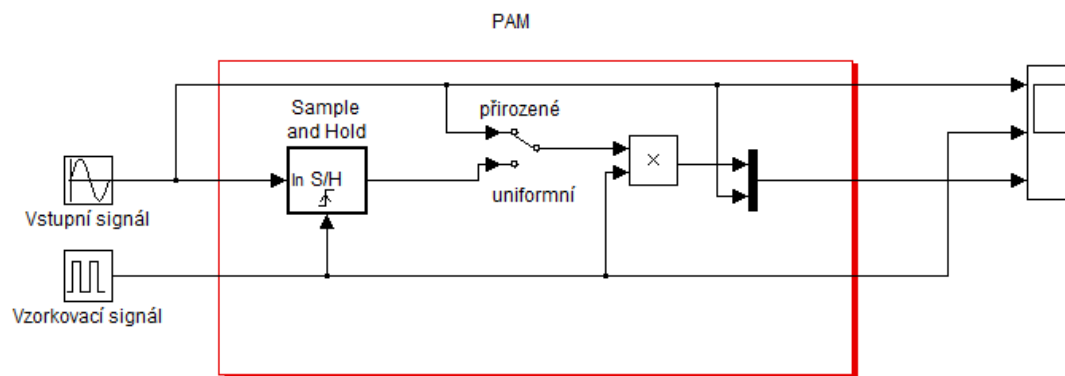
$$f_{VZ} \geq 2 \cdot f_{\max}, \quad (2)$$

kde $f_{\max}$ (Hz) je maximální frekvence obsažená ve vstupním signálu.

V praxi se však považuje za vhodnější vzorkovací kmitočet třikrát až pětikrát větší než maximální kmitočet obsažený v signálu. [3]

2.1 Pulzně amplitudová modulace (PAM)

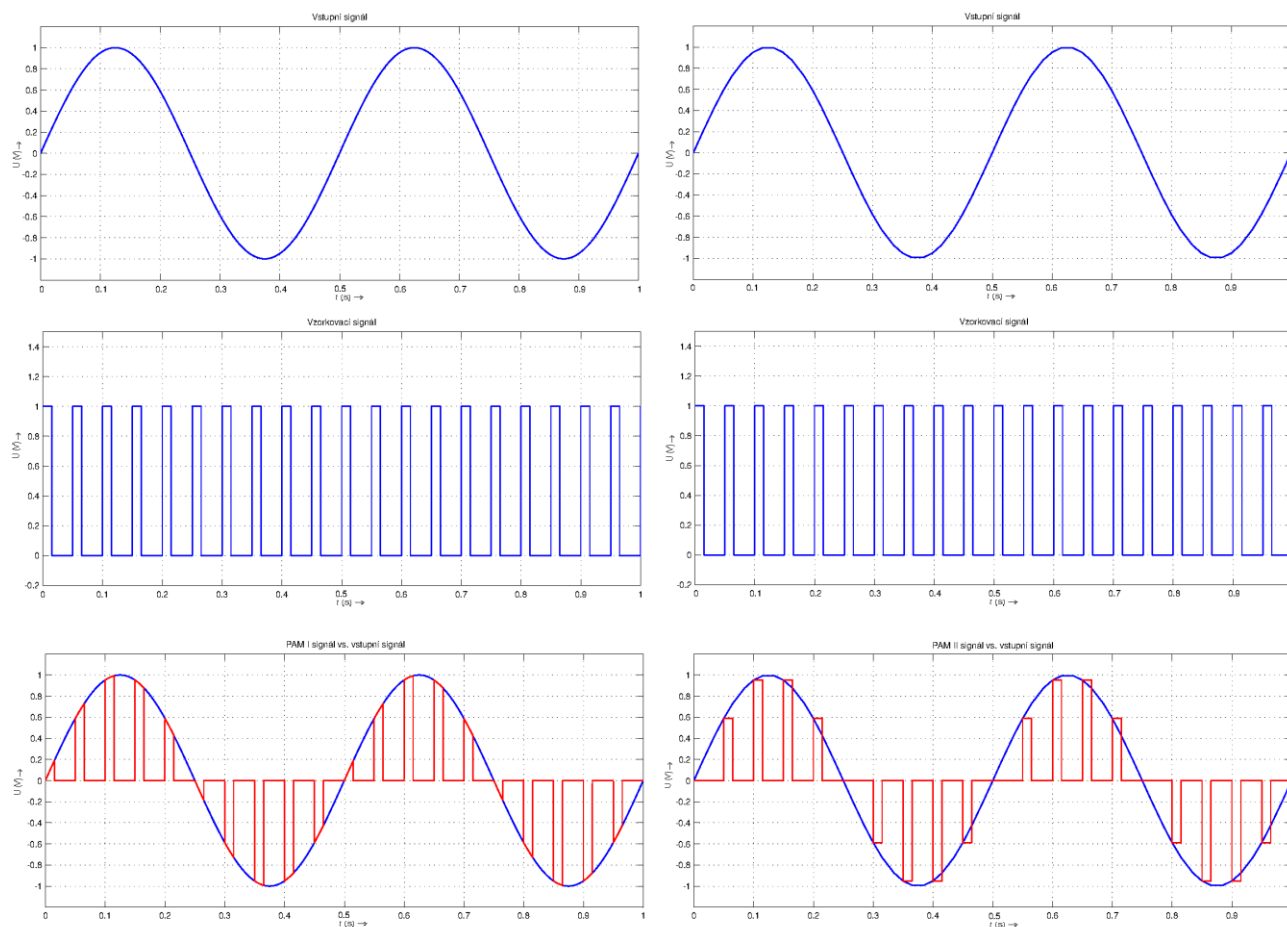
Jak vyplývá z názvu, jedná se o modulaci, při které je modulačním signálem ovlivňována výška (amplituda) impulzů. Jedná se tedy o obdobu klasické AM. V podstatě se jedná pouze o vzorkování. Právě podle toho, jaký druh vzorkování se zvolí, dostáváme dva druhy PAM. Při použití přirozeného vzorkování, dostáváme PAM I (1.druhu). V případě, že použijeme uniformní vzorkování, dostáváme PAM II (2.druhu). Zjednodušené schéma modulátoru PAM je zobrazeno na obrázku č.2.



Obrázek č.2: Zjednodušené blokové schéma modulátoru PAM

Jak je z tohoto obrázku patrné, skládá se zapojení pouze ze zdroje vstupního signálu, zdroje vzorkovacího obdélníkového signálu a vzorkovacího obvodu, ve kterém si lze zvolit druh vzorkování (přirozené/uniformní).

U přirozeného vzorkování je vstupní analogový signál násoben s jednotkovým signálem po dobu trvání vzorkovací periody T_{VZ} . Obvod *Sample and Hold* je tedy úplně vynechán. U uniformního (rovnoměrného) vzorkování, je ze vstupního analogového signálu nejprve odebrán vzorek, jehož hodnota je po dobu T_{VZ} stálá a je násobena, podobně jako v případě přirozeného vzorkování, jednotkovým signálem. Na obrázku č.3, jsou graficky naznačeny oba zmíněné principy pulzně amplitudové modulace.



Obrázek č.3: PAM 1. a 2.druhu

Jak je patrné z obrázku č.3, u pulzně amplitudové modulace prvního druhu modulovaný signál sleduje v době trvání impulsů průběh modulačního signálu. Impulzy modulovaného signálu u

pulzně amplitudové modulace druhého druhu se jeví z obrázku jako obdélníkové, což je však dáno pouze simulací, ve skutečnosti nejsou obecně obdélníkové, díky sebemenšímu zkreslení na obvodu sample and hold. [1]

Pro další uvažování je však tato skutečnost zanedbána a z důvodu zjednodušení bude PAM signál druhého druhu považován za obdélníkový.

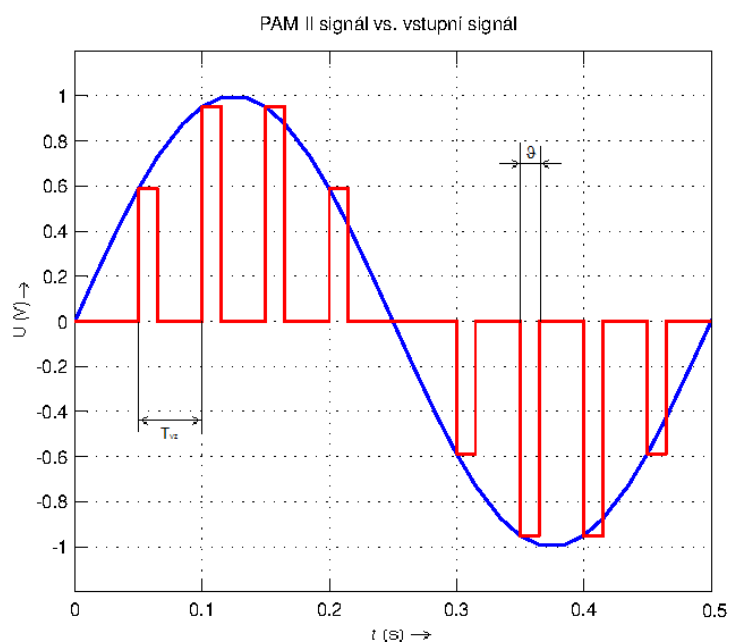
V dalších částech této kapitoly bude uvažována pouze PAM II.

Důležité parametry výstupního signálu jsou zobrazeny na obrázku č.4 a uvedeny v tabulce č.1

Tabulka č.2.1: Důležité parametry PAM II signálu

Označení parametru	Význam parametru
D	Výška vzorkovacího pulzu
ϑ	Šířka pulzu
T_{VZ}	Vzorkovací perioda

Parametry T_{VZ} a ϑ jsou vyznačeny na obrázku č.4, parametr D , zde zaznačen není z důvodu přehlednosti. Jak již však bylo zmíněno v předešlém textu, vzorkovací signál se vždy skládá z impulzů o velikosti D , s dobou trvání ϑ a se vzorkovací periodou T_{VZ} .



Obrázek č.4: Zobrazení důležitých parametrů PAM II signálu

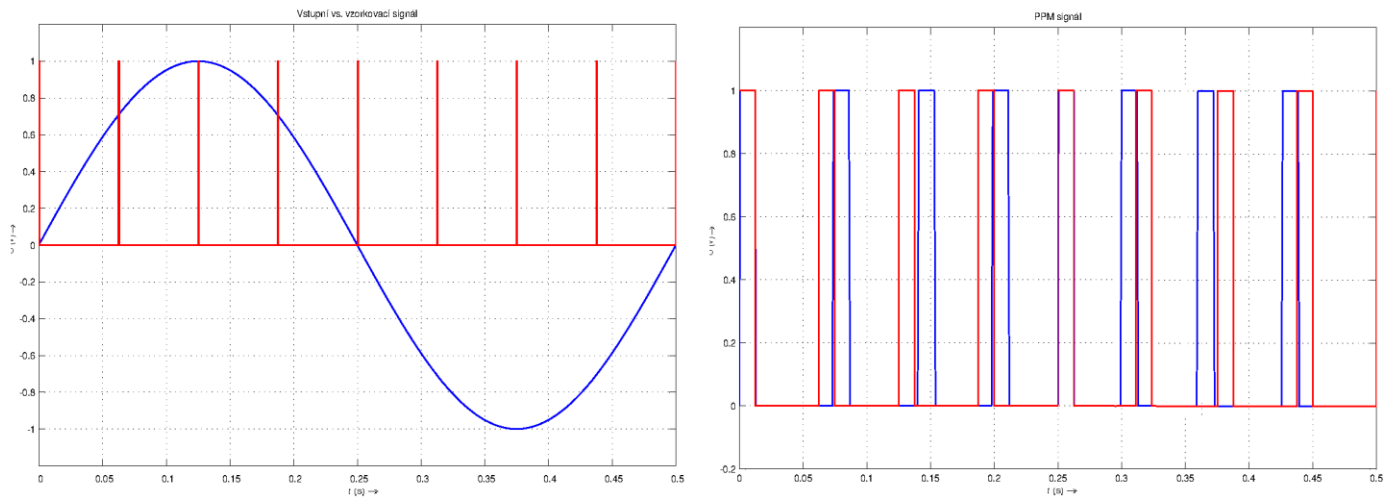
Protože se jedná o amplitudovou modulaci, je tento druh náchylný na rušení a zkreslení, není tedy vhodná pro přenos informací. Její hlavní použití je v dalších systémech, například v modulaci PCM, jak bude uvedeno dále. [3]

2.2 Pulzně polohová modulace (PPM)

Další z nekvantovaných impulzových modulací je Pulzně polohová modulace. U této modulace je okamžitá hodnota modulačního signálu vyjádřena časovou polohou impulsu v daném rámci, jehož délka zpravidla odpovídá vzorkovací periodě. [4]

Pulzně polohová modulace je obdobou fázové modulace PM.

Na obrázku č.5 jsou znázorněny časové průběhy signálů při PPM modulaci.



Obrázek č.5: Časové průběhy u modulace PPM

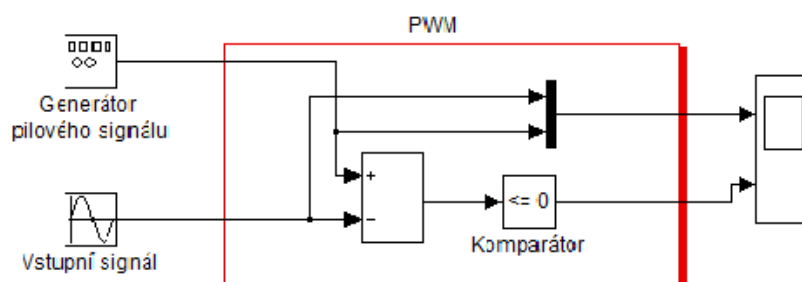
Na obrázku výše jsou znázorněny časové průběhy signálů použitých při PPM. Na levém je zobrazen modrou čarou vstupní analogový signál a červenou čarou vzorkovací signál. Na pravém obrázku jsou červenou čarou znázorněny výstupní impulzy PPM signálu v jejich poloze, která odpovídá vstupnímu signálu s hodnotou 0V. Modrou čarou jsou znázorněny výstupní impulzy, odpovídající vstupnímu signálu.

Princip PPM je ve své podstatě velmi jednoduchý. Je-li hodnota vstupního signálu vyšší než rozhodovací úroveň, posouvají se pulzy výstupního signálu o hodnotu přímo úměrnou aktuální hodnotě vstupního signálu oproti své obvyklé pozici doprava, tedy zpožďují se. Naopak, je-li vstupní signál nižší než rozhodovací úroveň, posouvají se pulzy výstupního signálu doleva oproti své obvyklé poloze, dá se tedy říci, že se předbíhají. [3]

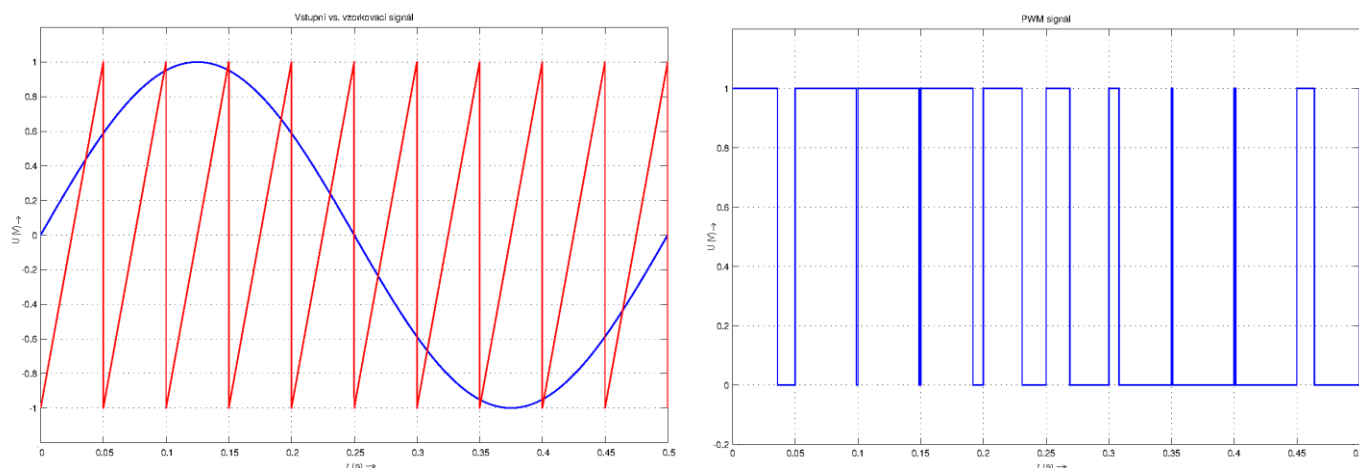
2.3 Pulzně šířková modulace (PWM)

Poslední nekvantovanou impulzní modulací, o které bude práce pojednávat, je Pulzně šířková modulace. Jak vyplývá z názvu, jedná se o modulaci, při níž je modulačním signálem ovlivňována šířka impulzů. Na rozdíl od PAM a PPM, nemá tato modulace obdobu u klasických analogových modulací.

Podobně jako v případě PAM se jedná pouze o vzorkování, ovšem s tím rozdílem, že vzorkovací signál není obdélníkový, ale má pilový průběh. Výstupní PWM signál je obvykle ve tvaru NRZ unipolárního signálu. Princip PWM je naznačen na obrázku č.7



Obrázek č.6: Zjednodušené blokové schéma PWM



Obrázek č.7: Časové průběhy PWM

Jak je z tohoto obrázku patrné, modulace je uskutečňována jako vzorkování vstupního signálu signálem pilovým. V okamžiku, kdy je hodnota vstupního analogového signálu vyšší nebo rovna, než hodnota pilového vzorkovacího signálu, překlopí komparátor svůj výstup tak, že výstupní PWM signál se nachází v logické úrovni „H“. Naopak, pokud nastane situace, že hodnota pilového vzorkovacího signálu je vyšší než hodnota vstupního signálu, dojde k překlopení výstupního signálu do logické úrovně „L“.

Pulzně šířková modulace je odolnější než pulzně amplitudová modulace, s čehož také vyplývá použití. Velmi často se využívá ve výkonové elektronice pro řízení DC/DC měničů, nebo řízení chodu elektromotorů. [1]

3. Kvantované modulace

Po části, věnující se nekvantovaným modulacím, následuje část věnující se kvantovaným impulzovým modulacím.

Jak bylo zmíněno výše, jedná se o jeden ze dvou základních druhů impulzových modulací. Princip tohoto druhu impulzových modulací se obvykle skládá ze tří postupných kroků. Proto je vhodné uvést jejich principy, aby nebylo nutné je popisovat později. Těmito kroky jsou:

- Vzorkování
- Kvantování
- Kódování

Princip vzorkování byl popsán již v kapitole pojednávající o nekvantovaných modulacích, proto mu nebude věnována větší pozornost.

Kvantování

Kvantování je operace, při které se každému vzorku přiděluje jedna z kvantizačních hladin. Jejich počet je dán počtem bitů výstupního signálu. Podle vztahu (3) je počet kvantizačních hladin definován jako exponenciální funkce ve tvaru

$$N_{KV} = 2^{nb}, \quad (3)$$

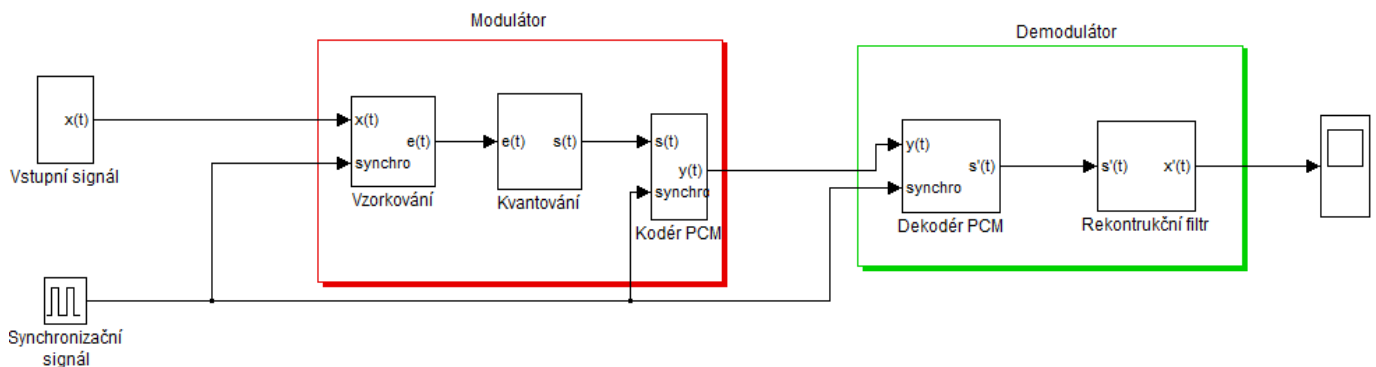
kde $N_{KV}(-)$ je počet kvantizačních hladin a $nb(-)$ je počet bitů výstupního signálu.

Kódování

Kódování je proces, který provádí konverzi kvantovaného signálu na výstupní signál modulátoru. Výstupní signál může mít pro jednotlivé modulace různý tvar. Například u PCM/DPCM může být výstupní signál ve tvaru unipolárního RZ signálu. Na rozdíl od toho, při použití DM a jejich alternativ, je nejčastěji využíván bipolární RZ signál.

3.1 Pulzní kódová modulace (PCM)

Jak již samotný název napovídá, provádí tato modulace převod analogového vstupního signálu na výstupní kódovou posloupnost pulzů. Tyto pulzy nejčastěji tvoří, jak již bylo zmíněno v předchozím odstavci, unipolární RZ signál.



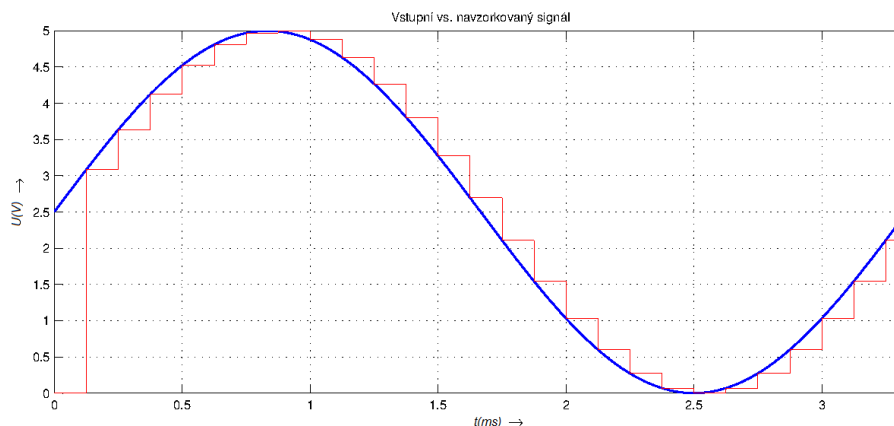
Obrázek č.8: Zjednodušené blokové schéma PCM

Na obrázku č.8 je znázorněno zjednodušené blokové schéma modulátoru a demodulátoru PCM. Jak je na tomto obrázku vidět, proces modulace PCM se skládá ze tří výše vysvětlených kroků.

- Vzorkování – rozklad na jednotlivé časové úseky
- Kvantování – rozdělení na úrovně podle úrovně signálu
- Kódování – překlad jednotlivých kvantizačních úrovní na číselnou (binární) podobu

V následujících několika odstavcích bude popsán návrh demonstračního modelu v prostředí Matlab-Simulink. Celý bude uvažovat 8bitovou PCM se vzorkovací frekvencí 8kHz.

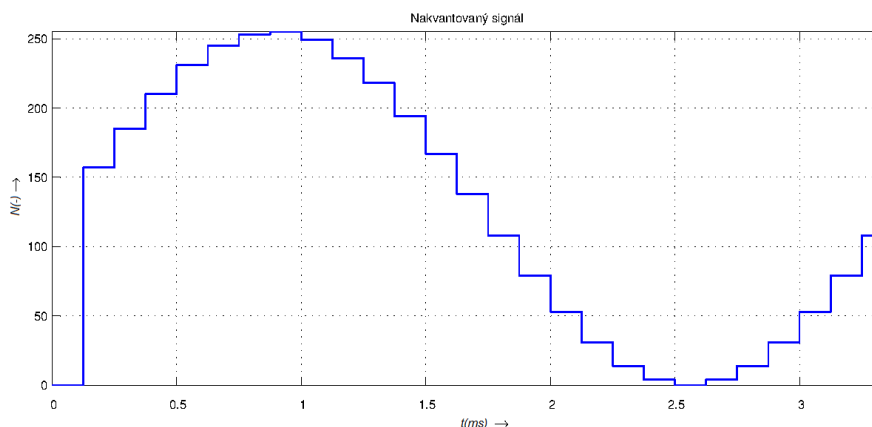
Vzorkování u PCM je provedeno pomocí PAM II, u které je délka pulzu rovna vzorkovací periodě. Platí zde tedy $T_p = T_{VZ}$. Tento PAM II signál je zobrazen na obrázku č.9 červeným průběhem. Modrou barvou je znázorněn vstupní analogový signál.



Obrázek č.9: Časové průběhy v modulátoru PCM

Dalším krokem PCM je **kvantování**. Jak je popsáno v jednom z předchozích odstavců, jedná se o přiřazení jednotlivých vzorků jedné z kvantizačních hladin. S využitím vztahu (3)

se určí počet kvantizačních hladin pro nbitovou PCM. V tomto případě se jedná o 8bitovou PCM, proto je počet kvantizačních hladin 256. Časový průběh nakvantovaného signálu je zobrazen na obrázku č.10.



Obrázek č.10: Časový průběh nakvantovaného signálu

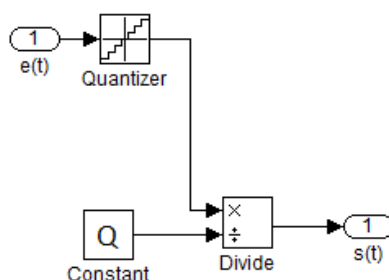
Jelikož vstupní signál nabývá hodnot v rozsahu 0V-5V, je nutné tuto škálu rozdělit právě na 256 úrovní. Kvantizační krok se vypočítá podle vztahu

$$q = \frac{FS}{N_{KV} - 1}, \quad (4)$$

kde $q(V)$ je kvantizační krok,
 $FS(V)$ je rozsah vstupního signálu a
 $N_{KV}(-)$ je počet kvantizačních hladin.

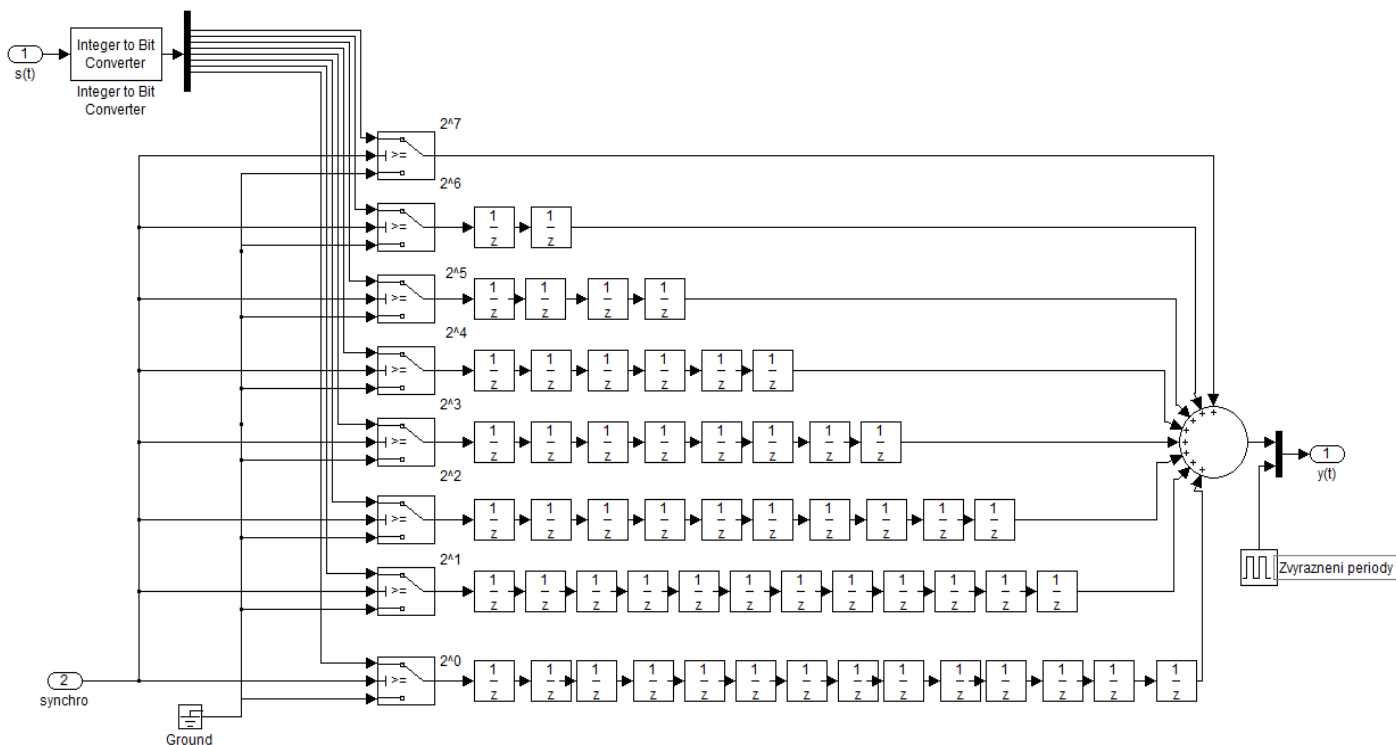
Je velmi důležité nezapomenout, že i 0V je jednou z kvantizačních úrovní, proto se tedy rozsah 5V nedělí počtem kvantizačních úrovní, nýbrž číslem, o jedna menším, v tomto případě tedy číslem 255.

Na obrázku č.11 je zobrazeno zapojení bloku kvantování. Skládá se z bloků *Quantizer*, *Divide* a bloku *Constant*. Vstupní signál do bloku kvantování, $e(t)$, je v rozsahu 0V-5V po průchodu blokem *Quantizer* je rozsah signálu stejný, tedy 0V-5V, ovšem tento rozsah je rozdělen do 256 úrovní. Aby bylo výstupem celého bloku kvantování, $s(t)$, celé číslo v rozsahu 0-255 je nutné provést podělení výstupu z bloku *Quantizer* v bloku *Divide* číslem q , což je kvantizační krok.



Obrázek č.11: Zapojení bloku kvantování

Po úspěšném provedení kvantování nastává poslední krok modulace, **kódování**. Tento krok se provádí v bloku *Kodér PCM*. Jeho vstupem, $s(t)$, je celočíselná hodnota v rozsahu 0-255 a požadovaným výstupem je sekvence pulzů vyjadřující hodnotu $s(t)$ v binárním kódu. Tato operace se provede za použití *bufferu*. Na obrázku č.12 je naznačeno zapojení kodéru. Jak je na tomto obrázku vidět, do kodéru vstupuje nejen celočíselný signál $s(t)$, ale také synchronizační signál, který je velmi důležitý pro bezproblémové kódování. Výstupem je již zmiňovaná posloupnost pulzů, vyjadřující celočíselnou hodnotu vstupu $s(t)$ v binárním kódu. K této posloupnosti je ještě pro zvýraznění periody přidán obdélníkový signál.



Obrázek č.12: Blokové schéma zapojení kodéru

Jádrem celého bloku kodéru je však mechanismus vyčítání hodnot ze vstupního signálu $s(t)$ a jejich plnění, do rámce výstupního signálu $y(t)$. Tento mechanismus je složen z osmi bloků *Switch*, které jsou řízeny synchronizačním signálem. Synchronizační signál má obdélníkový průběh a doba trvání impulsu je rovna hodnotě vypočítané podle vztahu (5).

$$t_b = \frac{T_v}{2 \cdot nb}, \quad (5)$$

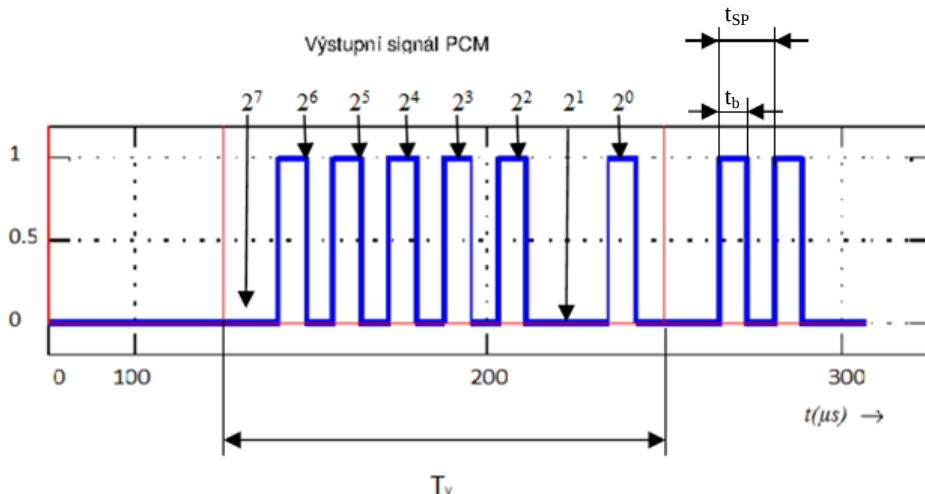
kde $t_b(s)$ je doba trvání synchronizačního impulsu,

$T_v(s)$ je vzorkovací perioda a

$nb(-)$ je počet bitů výstupního signálu.

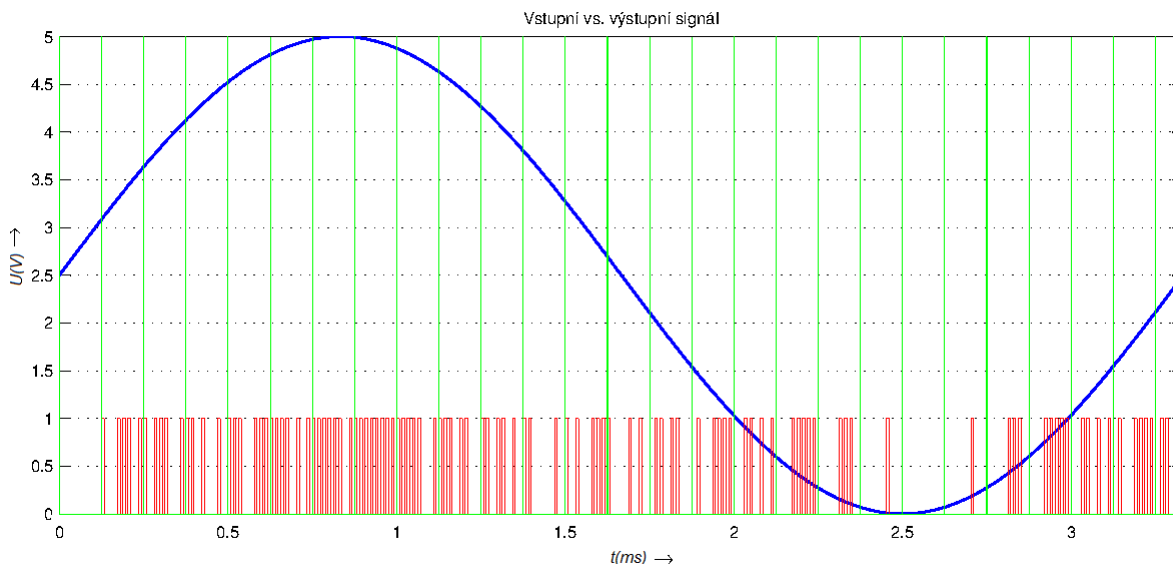
Pro 8bitovou PCM se vzorkovacím kmitočtem 8kHz je doba trvání synchronizačního impulsu rovna 7,8125μs. Každou periodu se nastaví výstup jednotlivých bloků *switch* na hodnotu odpovídající jednotlivým bitům. O to aby byly tyto bity poslány na výstup kodéru ve správný čas, se stará 8 zpožďovacích linek. Zpožďovací linka pro signálový prvek reprezentující hodnotu 2^7 má hodnotu zpoždění 0μs, protože tento signálový prvek je poslán na výstup kodéru jako první. Zpoždění druhé zpožďovací linky je nastaveno na hodnotu $2 \cdot t_b$, jelikož signálový prvek reprezentující hodnotu 2^6 je vyslán na linku jako druhý. Každá další zpožďovací linka má zpoždění o $2 \cdot t_b$ větší.

Jak již bylo několikrát zmíněno, výstupní zakódovaný signál je ve tvaru unipolární RZ posloupnosti pulzů. Doba trvání signálového prvku (t_{SP}), je vždy dvojnásobně dlouhá než doba trvání synchronizačního impulzu (t_p). První část signálového prvku je tvořena hodnotou odpovídajícího bitu na výstupu bloku *Integer to Bit Converter*, druhá část je vždy rovna nule. Příklad výstupního signálu PCM s ukázkou jednotlivých signálových prvků je zobrazen na obrázku č.13.



Obrázek č.13: Výstupní signál PCM

Časové průběhy vstupního (modrý), synchronizačního(zelený) a výstupního signálu PCM jsou naznačeny na obrázku č.14.



Obrázek č.14: Časové průběhy PCM

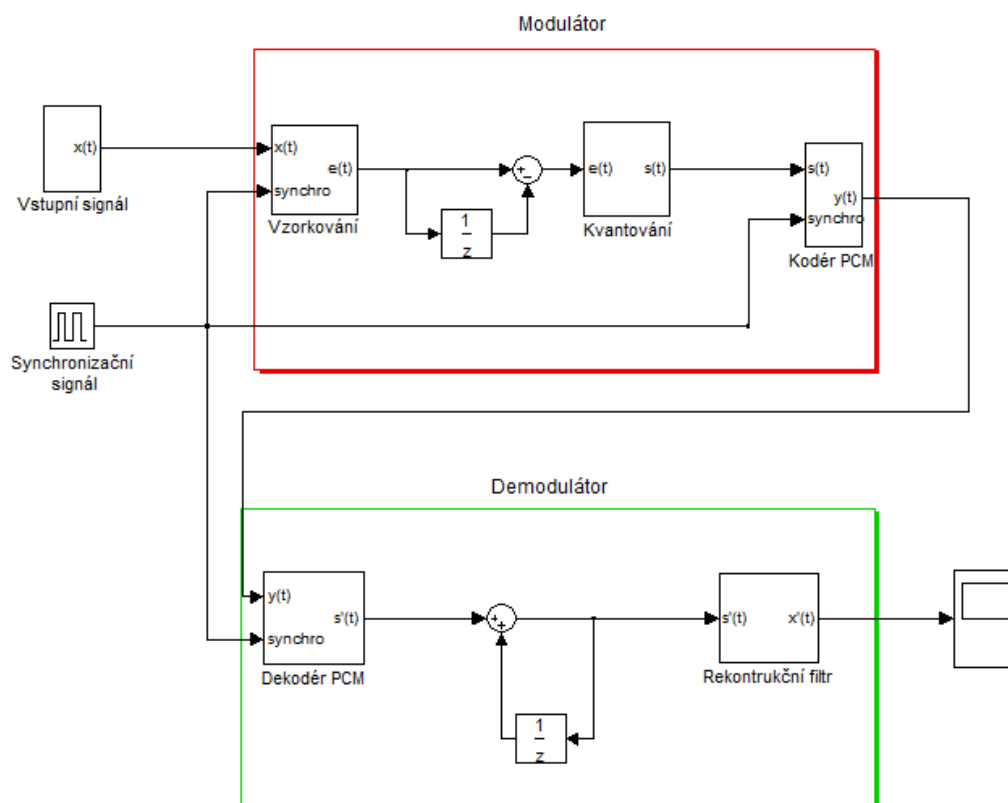
Pulzně kódová modulace je velmi často používána jako základ převodníků D/A a A/D. Je možné se s ní potkat také u zdrojového kódování v telekomunikačním řetězci, kde nejčastěji plní právě roli A/D převodníku.

3.2 Diferenční pulzní kódová modulace (DPCM)

Při vzorkování elektroakustických signálů, obrazových signálů (ale i u dalších signálů) existuje mezi jejich po sobě následujícími vzorky výrazná korelace. Jak je popsáno v předcházející kapitole, týkající se PCM, jsou tyto vzorky kvantovány a poté přímo kódovány. Přenáší se tedy skutečné velikosti kvantovaných vzorků modulačního signálu.

Vezmeme-li v potaz informaci napsanou na začátku tohoto odstavce, tedy že u většiny reálných signálů jsou sousední vzorky na sobě závislé a jejich rozdíl je menší než samotné vzorky, je vhodnější kvantovat pouze rozdíly mezi sousedními vzorky.[1]

U diferenční pulzní kódové modulace (dále pouze DPCM) se tedy nepřenáší informace o skutečné velikosti kvantovaného vzorku modulačního signálu, ale o velikosti kvantovaného rozdílu tohoto vzorku a jeho předpokládané hodnoty, odvozené na základě analýzy několika předcházejících vzorků. Pokud má vstupní signál analogový signál takový charakter, že je možné jej spolehlivě predikovat, jsou vzorky rozdílového signálu relativně velmi malé vůči vzorkům modulačního signálu. Tím jsou k jejich přenosu potom zapotřebí kratší kódové skupiny a také menší šířka pásma, než je vyžadována u modulace PCM, neboť se nepřenáší nadbytečná informace.[1]



Obrázek č.15: Zjednodušené blokové schéma DPCM

Na obrázku č.15 je zobrazeno zjednodušené blokové schéma modulátoru a demodulátoru DPCM. Jelikož je DPCM pouze modifikací PCM, jsou si obě schémata velmi podobná. Zásadním rozdílem je blok provádějící rozdíl mezi dvěma po sobě jdoucími vzorky u modulátoru. U demodulátoru je zde navíc blok, který k aktuálnímu přenesenému vzorku přičítá hodnotu vzorku přeneseného v minulém kroku.

Podobně jako v části věnované PCM, bude několika následujících odstavcích popsán návrh demonstračního modelu v prostředí Matlab-Simulink. Stejně jako v případě PCM, bude i u DPCM uvažována 8bitová DPCM a vzorkovací kmitočet 8kHz.

Vzorkování a kódování je u DPCM provedeno stejně jako u PCM. Mírně se však liší blok provádějící kvantování.

Protože při rozdílu dvou po sobě jdoucích vzorků může nastat, že rozdíl bude záporný, a kodér, jak bylo ukázáno v minulé kapitole, dokáže zakódovat pouze celá čísla v rozsahu 0 až $2^{nb}-1$, je nutné ošetřit tuto situaci například tak, že se bude kvantovat místo na 8 bitů pouze na 7 bitů. Osmý bit bude sloužit k posunutí signálu z oblasti záporných čísel. Podle vztahu (3) je počet kvantizačních úrovní pro 7bitovou modulaci PCM 128. Podle vztahu (4) má kvantizační krok velikost 0,0196V.

Aby bylo možné provést změnu z 8 na 7 bitů, je nutné si ověřit, zda bude mít 7bitová DPCM stejné přenosové vlastnosti jako 8bitová PCM. V úvahu je brán vstupní analogový signál s amplitudou 2,5V a s maximální frekvencí 500Hz. Vzorkovací kmitočet je 8000Hz.

Podle vztahu (6) je vypočítána maximální difference vstupního signálu.[4]

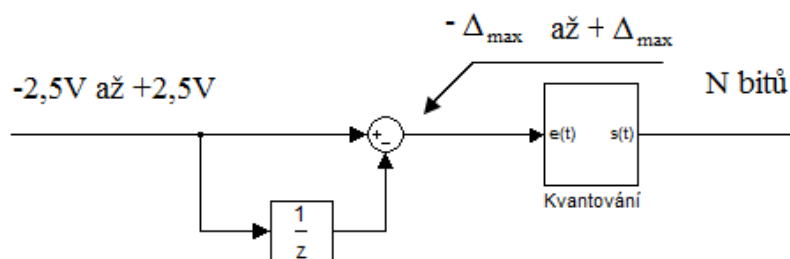
$$\Delta_{\max} = 2 \cdot A \cdot \sin\left(2 \cdot \pi \cdot f_{\max} \cdot \frac{1}{2 \cdot f_{VZ}}\right), \quad (6)$$

kde $\Delta_{\max}(V)$ je maximální difference vstupního signálu,

A je amplituda vstupního sinusového signálu a

$f_{\max}(Hz)$ je maximální kmitočet obsažený v signálu.

Pro dané hodnoty je tato maximální difference rovna 1,248V. To znamená, že signál vstupující do bloku *Kvantování* má maximální rozsah $\pm 1,248V$. Pro lepší představu je na následujícím obrázku vyznačeno místo ve schématu, o kterém se celou dobu jedná.



Obrázek č.16: Objasnění místa výskytu s maximální diferencí [4]

Počet bitů potřebných pro modulaci signálu je dán vztahem [4]

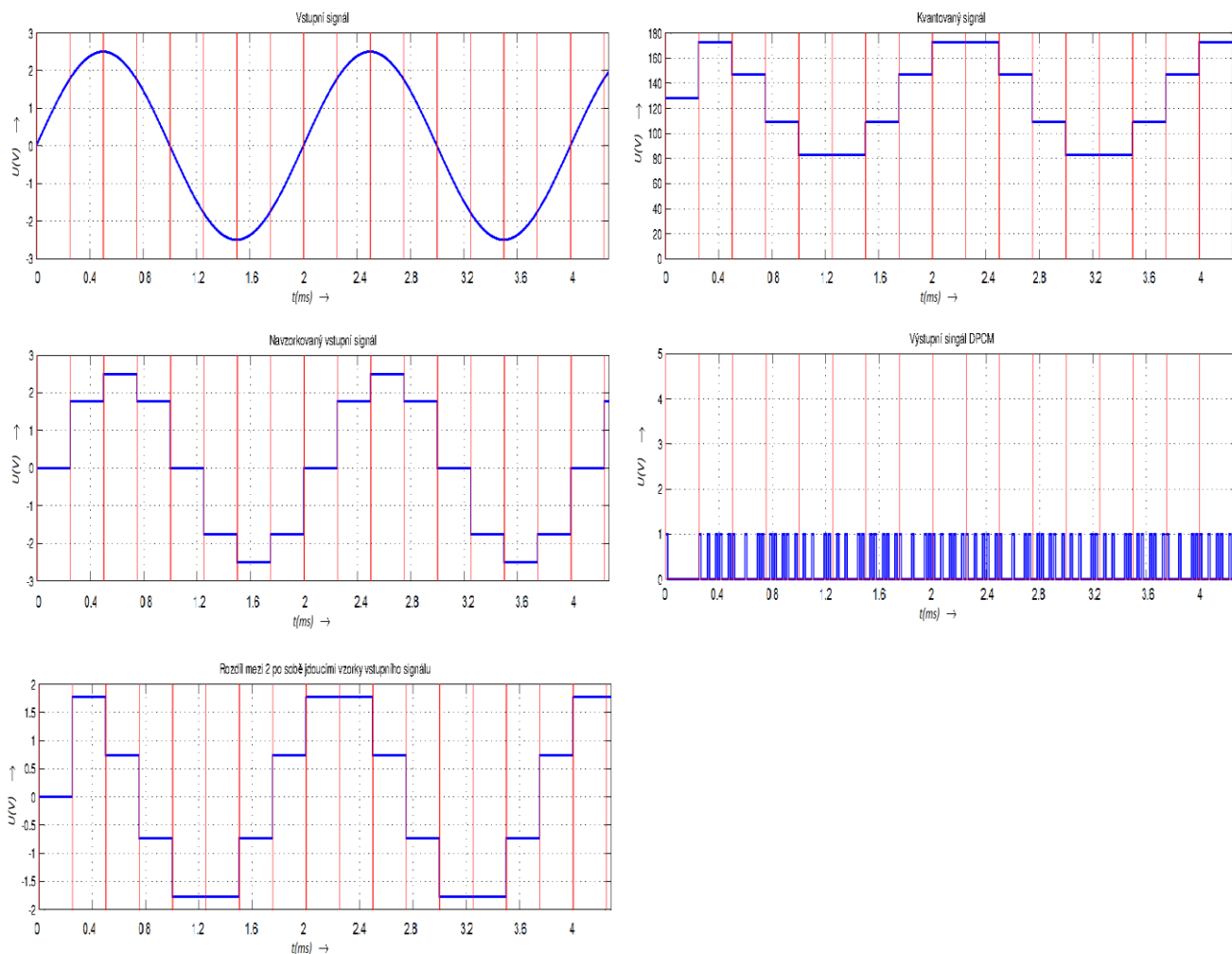
$$N = \log_2\left(\frac{2 \cdot \Delta_{\max}}{q}\right), \quad (7)$$

kde $N(-)$ je počet bitů a

$q(V)$ je kvantizační krok.

Pro výše vypočítanou maximální diferenci je počet bitů potřebných pro modulaci signálu roven 6,99. Protože ovšem nelze kvantovat na desetiny bitu, je počet bitů vždy zaokrouhlen nahoru, tedy na 7.

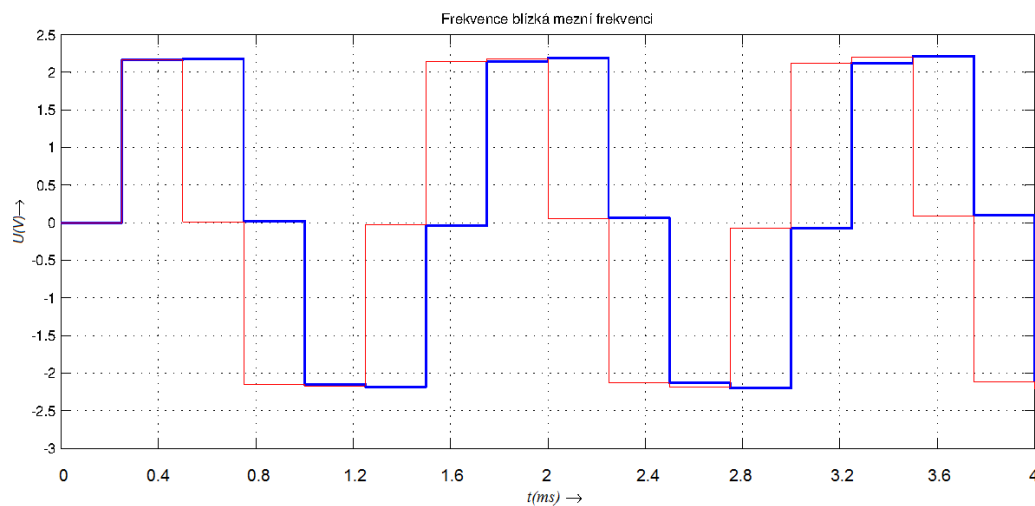
Touto krátkou úvahou bylo dokázáno, že 8bitovou PCM lze nahradit pro daný vstupní signál 7bitovou DPCM.



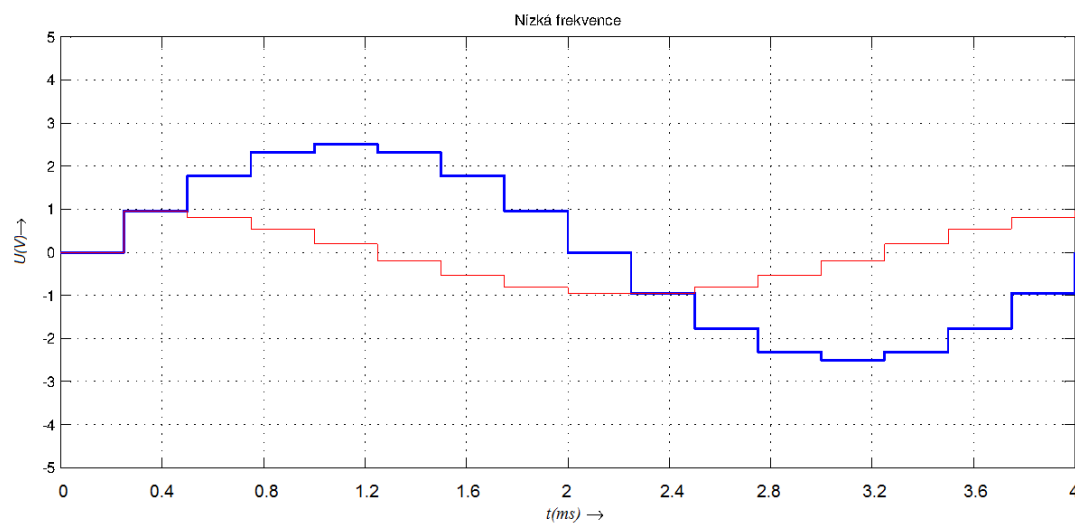
Obrázek č.17: Časové průběhy DPCM

Na obrázku č.17 jsou zobrazeny časové průběhy v modulátoru DPCM. Z něj je patrné, že kvantovaný signál nevyužívá celého rozsahu. To je dáno tím, že frekvence vstupního signálu není maximální možná, ale je pouze 500Hz. V případě, že by vstupní signál měl maximální možnou frekvenci, byly by rozdíly vstupních vzorků větší, tudíž by i kvantovaný signál využíval více svůj maximální rozsah.

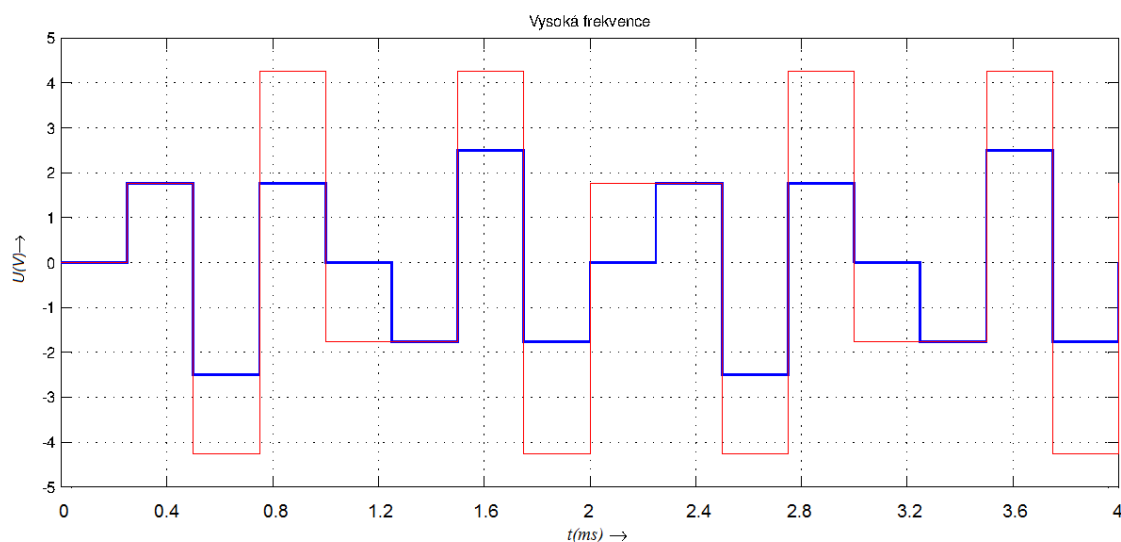
Na obrázcích č. 18, 19 a 20 jsou zobrazeny časové průběhy ilustrující možné případy, ke kterým může dojít v závislosti na volbě frekvence vstupního signálu. Modrou barvou je znázorněn časový průběh navzorkovaného signálu, červenou barvou časový průběh signálu rozdílového.



Obrázek č.18: Časové průběhy pro vstupní signál s frekvencí blízkou mezní frekvenci



Obrázek č.19: Časové průběhy pro vstupní signál s frekvencí nižší než mezní



Obrázek č.20: časové průběhy pro vstupní signál s frekvencí vyšší než mezní

Jak je patrné z obrázku č.18, při volbě vstupního signálu s maximální frekvencí blízkou mezní frekvenci, je maximální rozkmit rozdílového signálu téměř stejný jako rozkmit navzorkovaného signálu. Zda je tedy možné aby PCM byla nahrazena DPCM je nutné ověřit výpočtem.

Z obrázku č.19 je patrné, že při volbě vstupního signálu s maximální frekvencí nižší než mezní frekvencí, dochází k rapidnímu snížení maximálního rozkmitu rozdílového signálu. Díky tomu je vhodné využít DPCM namísto PCM.

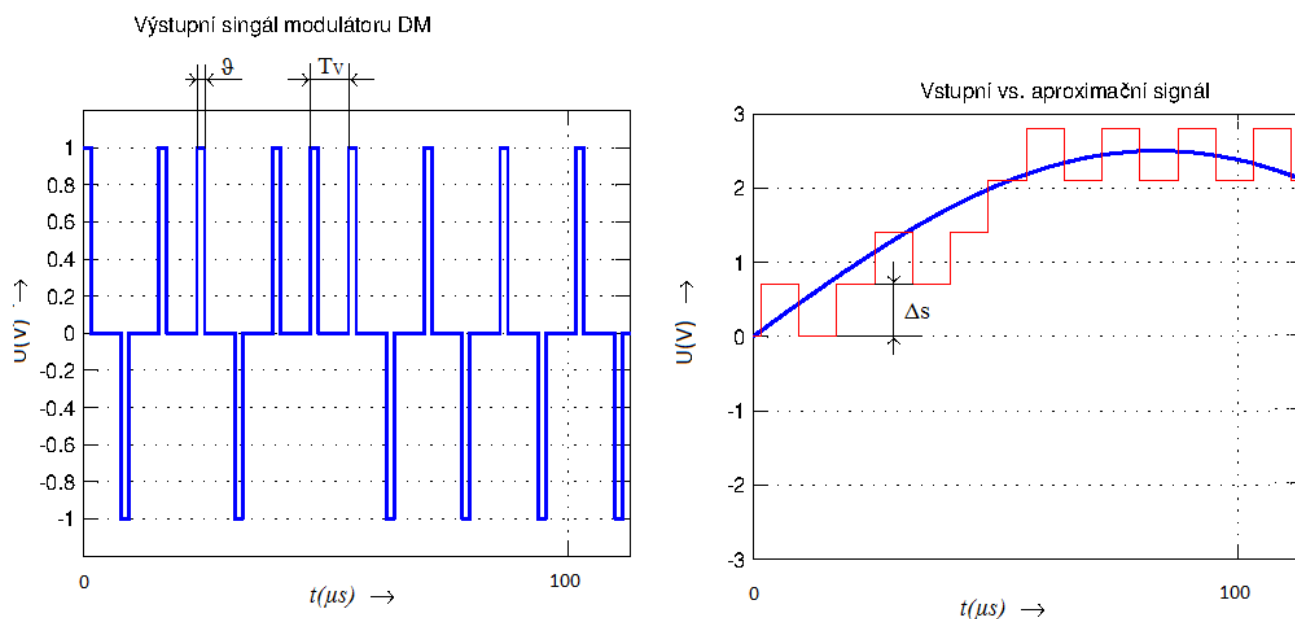
Na obrázku č.20 jsou zachyceny časové průběhy, které vzniknou při volbě vstupního signálu s maximální frekvencí vyšší, než je mezní frekvence. Díky tomu je rozkmit rozdílového signálu větší než rozkmit signálu navzorkovaného.

Jelikož mezní frekvence závisí na frekvenci vzorkovací, je z tohoto závěru zřejmé, že pro bezproblémovou funkčnost DPCM je nutná vyšší vzorkovací frekvence než pro PCM.

3.3 Delta modulace (DM)

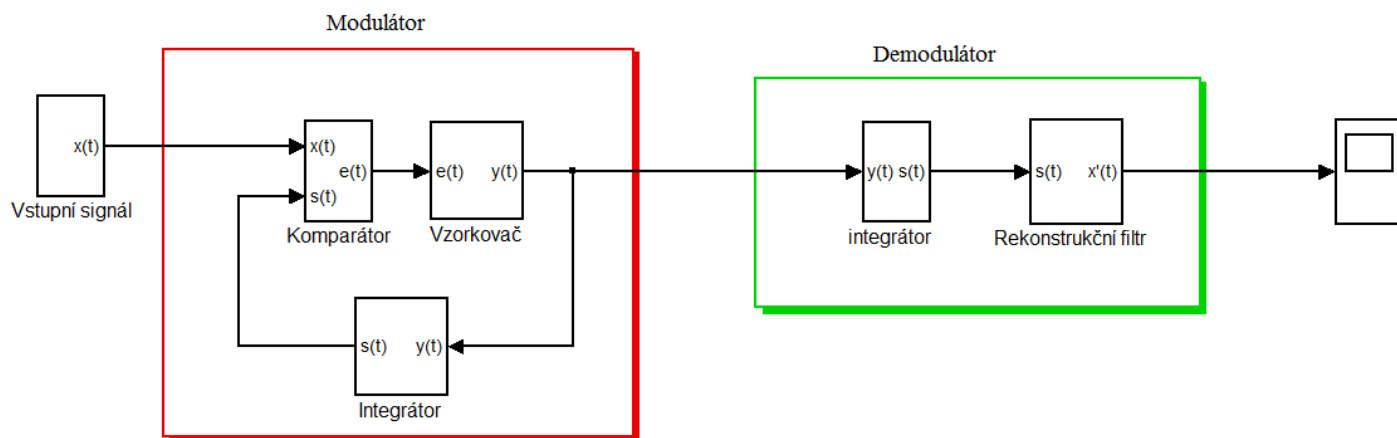
Princip delta modulace spočívá v porovnávání předchozího vzorku a aktuální hodnoty amplitudy přenášeného signálu. Pro každý vzorek, je přenášen pouze jeden binární signál, který určí, jestli je třeba na přijímací straně hodnotu amplitudy zvýšit nebo snížit oproti předchozímu přijatému vzorku. [1]

Na obrázku č.21 jsou zobrazeny důležité parametry používané při DM. Těmito parametry jsou vzorkovací perioda T_v , šířka výstupního impulzu θ a výška schodu Δs .



Obrázek č.21: Zobrazení důležitých parametrů DM

Delta modulace může být **lineární** nebo **adaptabilní**. U lineární DM, je výška aproximačních pulzů stále stejná. Jinými slovy se výška schodu nemění. V případě adaptabilní delta modulace se výška schodu mění. Více o této problematice bude zmíněno v kapitole ADM.



Obrázek č.22: Zjednodušené blokové schéma zapojení DM[4]

Jak je z tohoto obrázku patrné, skládá se modulátor DM z *komparátoru*, *vzorkovače* a *integrátoru* zapojeného ve zpětné vazbě. Demodulátor se skládá z *integrátoru* a *rekonstrukčního filtru*.

Vstupní signál $x(t)$ je v rozsahu $\pm 2,5V$. V bloku *Komparátor* je od tohoto signálu odečten signál $s(t)$, který je definován jako vstupní signál zpožděný o jednu periodu. Je-li tento rozdíl kladný, na výstupu komparátoru se nastaví signál o úrovni $1V$, pokud je rozdíl záporný, na výstupu komparátoru se nastaví signál o úrovni $-1V$. Signály s těmito úrovněmi vstupují do bloku *Vzorkovač*, kde se podle jejich úrovně nastavují výstupní pulzy modulátoru buďto na kladnou nebo na zápornou hodnotu. Tento výstupní signál jde kromě výstupu také do bloku *Integrátor*, kde se z něj vytváří aproximační signál.

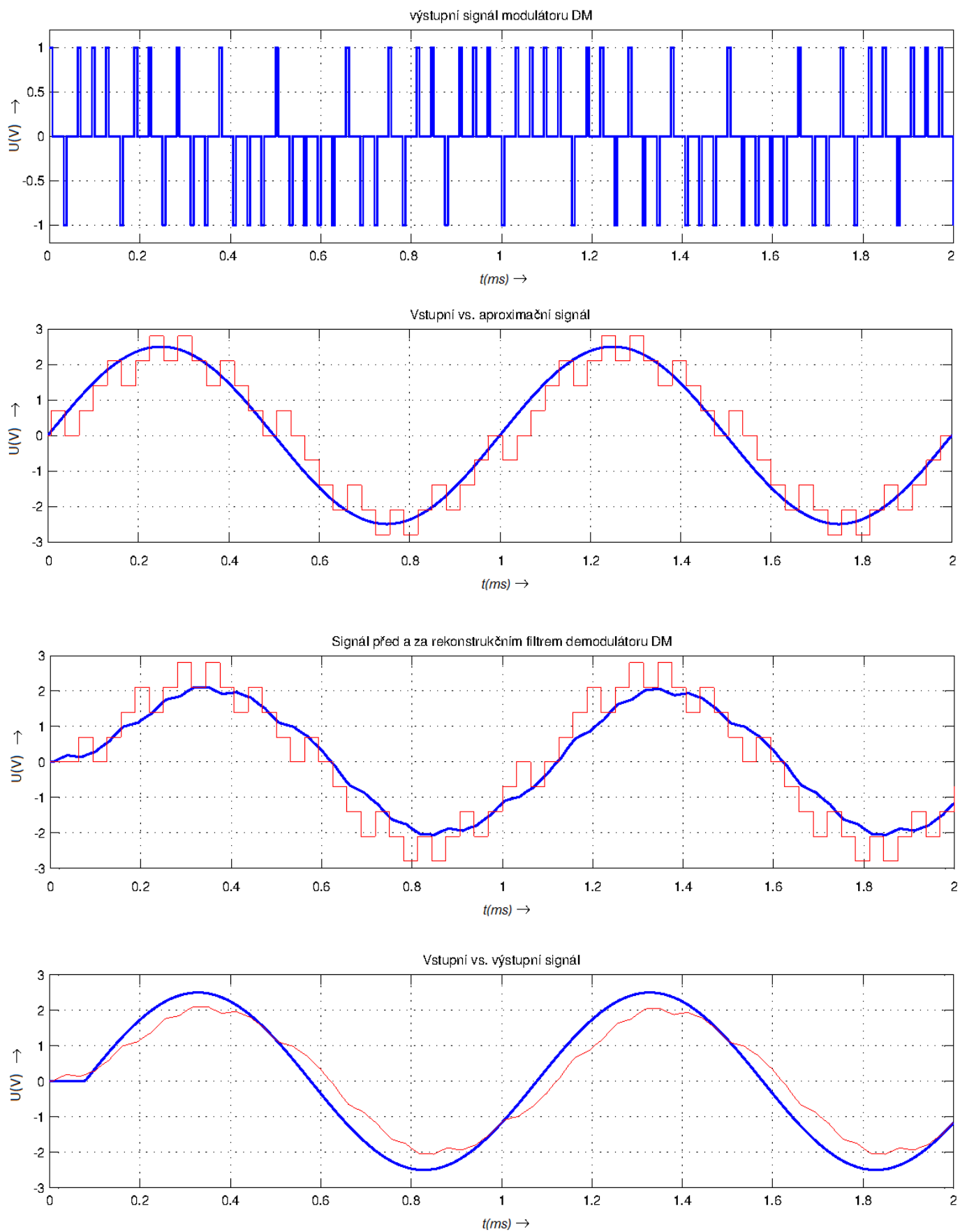
V demodulátoru signál prochází, podobně jako ve větvi zpětné vazby v modulátoru, blokem *Integrátor*, kde se z něj vytváří aproximační signál. Následujícím blokem je *rekonstrukční filtr*, který aproximovaný signál vyhladí do podoby vstupního signálu.

Na obrázku č.23 jsou zobrazeny časové průběhy signálů v modulátoru a demodulátoru. Na prvním obrázku je zobrazen výstupní signál modulátoru. Jedná se o bipolární RZ signál.

Na druhém obrázku je zobrazen vstupní analogový signál (modrý) a aproximační signál vystupující z *integrátoru* ve zpětné vazbě modulátoru DM (červený).

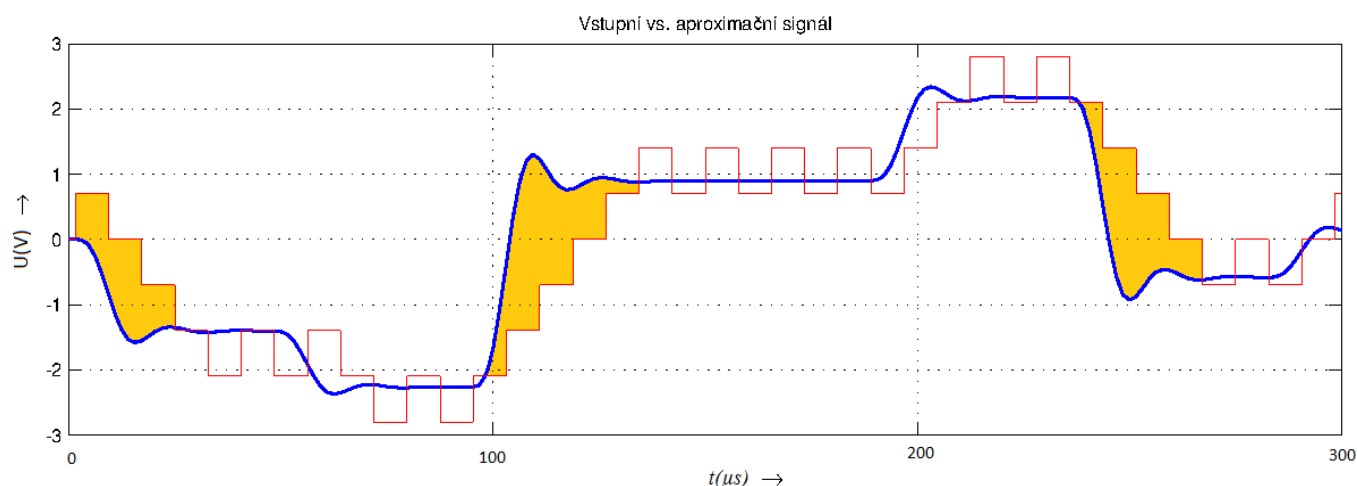
Na třetím obrázku by v ideálním případě měl být stejný průběh jako na druhém obrázku. Jedná se totiž o signál před *rekonstrukčním filtrem* (červený) a za *rekonstrukčním filtrem* (modrý) v demodulátoru DM.

Na posledním obrázku je časový průběh vstupního (modrý) a výstupního (červený) signálu. V ideálním případě by se měly průběhy překrývat, což ovšem není prakticky téměř možné z důvodu výpočetní náročnosti rekonstrukčního filtru.



Obrázek č.23: Časové průběhy signálů v modulátoru a demodulátoru DM

Velkou nevýhodou DM je náchylnost k tzv. **přetížení kodéru**. Jedná se o jev, kdy, aproximační signál nestíhá sledovat rychlé změny vstupního signálu. Zkreslení dané přetížením kodéru je zobrazeno na následujícím obrázku. [1]



Obrázek č.24: Zobrazení zkreslení dané přetížením kodéru

Aby nedocházelo k přetížení kodéru, je nutné zaručit maximální strmost vstupního signálu. Pro nejhorší případ platí: [4]

Vstupní signál: $x(t) = A \cdot \sin \Omega_{\max} t$

Maximální strmost: $\max[x'(t)] = A \cdot \Omega_{\max}$

Podmínka nepřetížení: $A \cdot \Omega_{\max} \leq \frac{\Delta s}{T_V}, \quad (8)$

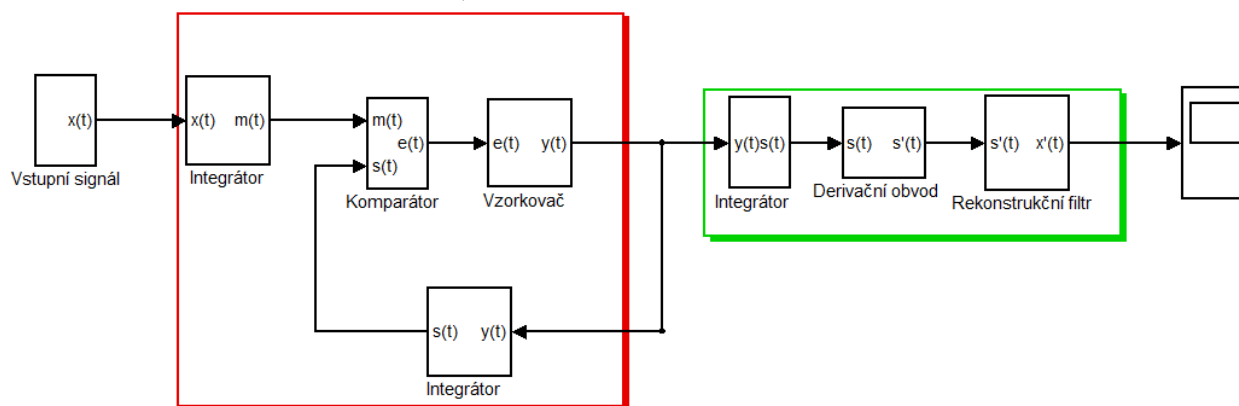
kde A je amplituda vstupního signálu,
 $\Omega_{\max}$ je maximální úhlový kmitočet vstupního signálu,
 Δs je výška schodu a
 T_V je vzorkovací perioda.

Pravděpodobnost, že dojde k přetížení kodéru u DM lze také snadno zmenšit úpravou modulačního signálu spočívající v jeho integraci. Doplněním integrátoru k modulátoru DM dostaneme modulátor Σ - Δ (Sigma-Delta). [4]

Další možností jak snížit pravděpodobnost přetížení kodéru je použití adaptabilní delta modulace (ADM).

3.4 Sigma-Delta modulace $\Sigma\text{-}\Delta$ (SDM)

Jak bylo uvedeno v předcházející kapitole, je tato modulace modifikací modulace delta, snižující pravděpodobnost přetížení kodéru. Toho lze dosáhnout integrací vstupního signálu. Schéma zapojení modulátoru a demodulátoru SDM je zobrazeno na obrázku č.25. Jedná se v podstatě o modulátor Delta doplněný o integrátor vstupního signálu. Demodulátor je doplněn o derivační obvod.[4]

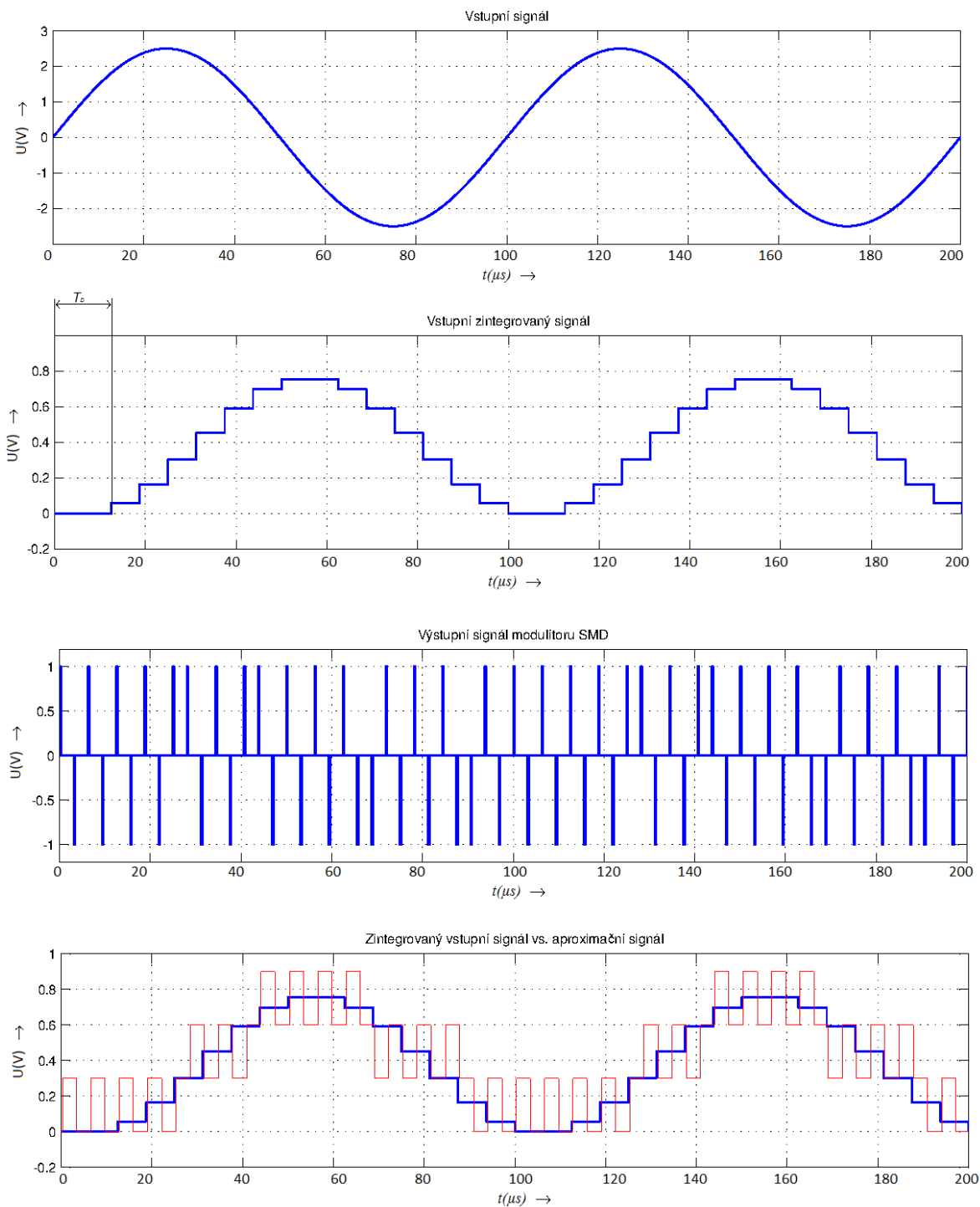


Obrázek č.25: Zjednodušené blokové schéma modulátoru a demodulátoru SMD [4]

Na obrázku č. 26 jsou zobrazeny časové průběhy signálů v modulátoru SMD.

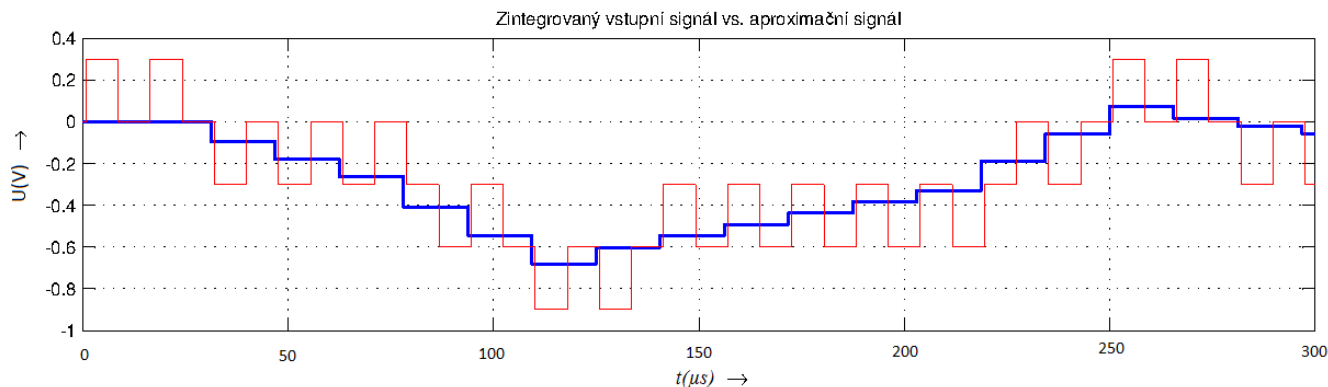
Na prvním obrázku je naznačen pouze časový průběh vstupního signálu, na druhém je zobrazen časový průběh zintegrovaného vstupního signálu. Na třetím obrázku je zobrazen časový průběh výstupního signálu modulátoru SDM. Na poslední obrázku je naznačen časový průběh vstupního zintegrovaného signálu a časový průběh aproximačního signálu.

Jak je z tohoto obrázku patrné, díky integraci je snížena dynamika vstupního signálu. Díky tomu je možné tento signál aproximovat i signálem s nižší výškou schodu, aniž by docházelo k přetížení kodéru. Tato skutečnost je zobrazena na obrázku č.27. Integrace ovšem do celého procesu modulace zanáší jisté zpoždění T_D . To je ukázáno na druhém časovém průběhu.



Obrázek č.26: Časové průběhy signálů v modulátoru SDM

Stejný signál, který byl použit u modulátoru Delta pro zobrazení zkreslení daného přetížením kodéru, byl použit také u modulátoru Sigma-Delta. Jak je patrné z následujícího obrázku, k přetížení kodéru nedošlo, přestože byla použita i nižší výška schodu.



Obrázek č.27: Absence zkreslení dané přetížením kodéru

Přestože, SDM snižuje pravděpodobnost vzniku zkreslení způsobené přetížením kodéru, mohou u této modulace vznikat další nepříjemnosti.

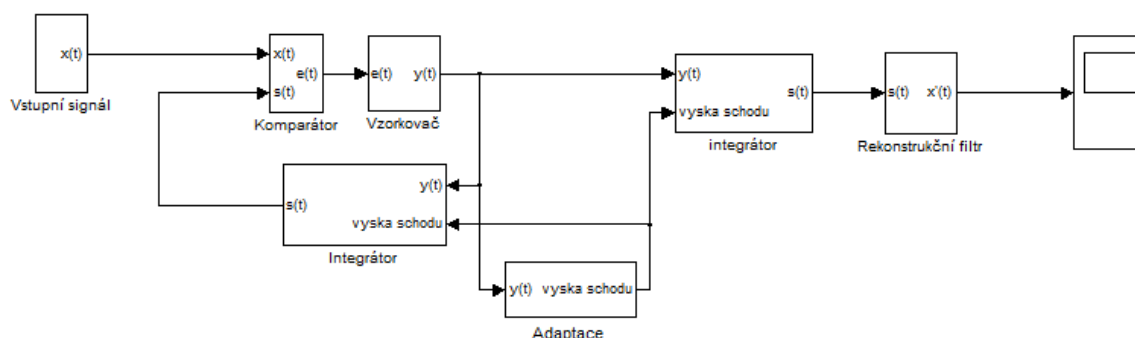
Za jednu z nich může být považováno již zmiňované zpoždění. Díky tomu není tento modulátor vhodný pro použití v reálném čase. [3]

3.5 Adaptabilní Delta modulace

Jako další způsob jak se vyhnout přetížení kodéru, je vhodné použít Adaptabilní Delta modulaci, u které již není výška schodu po celou dobu konstantní, nýbrž dochází k jejím změnám. Tomuto problému se věnovalo mnoho lidí a každý z nich vytvořil více či méně odlišný algoritmus, podle kterého se výška schodu mění. Příkladem může být například Winklerův algoritmus.[1]

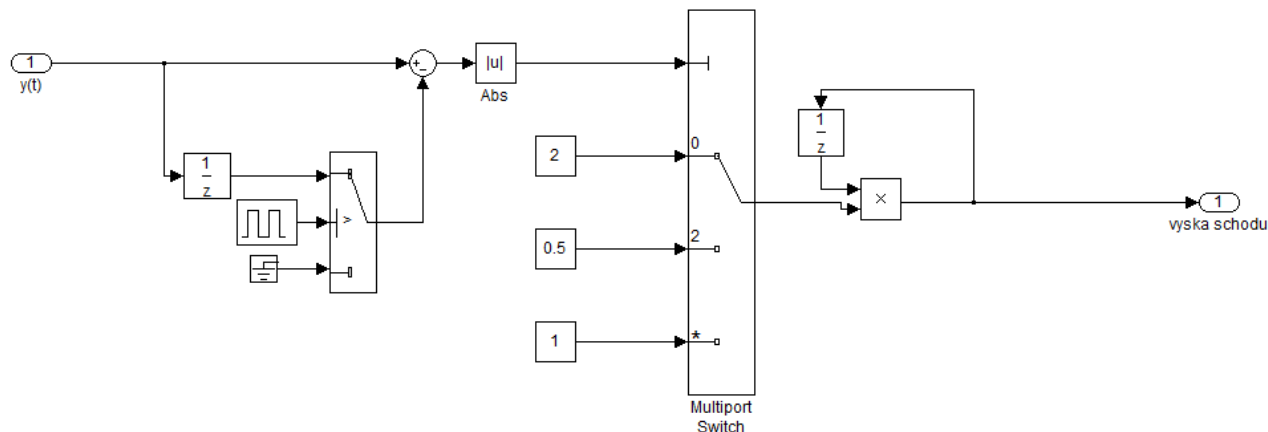
Winklerův algoritmus [3]

Jedná se o algoritmus, který mění výšku schodu podle výstupní posloupnosti modulátoru. V případě, že jsou si dva po sobě jdoucí pulzy rovny, zvýší se velikost schodu na dvojnásobek. V případě, že jsou dva po sobě jdoucí pulzy různé, velikost schodu se sníží na polovinu. Na obrázku č.28 je zobrazeno zjednodušené blokové schéma zapojení ADM s Winklerovým algoritmem.



Obrázek č.28: Zjednodušené blokové schéma modulátoru a demodulátoru ADM

Protože je ADM jednou z modifikací DM, je blokové schéma ADM velmi podobné blokovému schématu DM. Zásadním rozdílem je blok *Adaptace*, který se u klasické DM nevyskytuje. U ADM plní tento blok funkci změny velikosti schodu, což u klasické DM není za potřebí. Bloky *Komparátor* a *Vzorkovač* u modulátoru a blok *Rekonstrukční filtr* u demodulátoru jsou naprosto shodné s ekvivalentními bloky u DM. Blok *Integrátor* je mírně pozměněn. Zjednodušené blokové schéma bloku *Adaptace* je zobrazeno na obrázku č.29, Zjednodušené blokové schéma bloku *Integrátor* je zobrazeno na obrázku č.30.



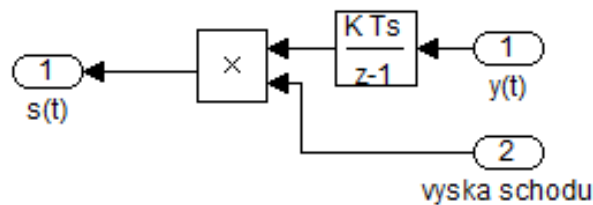
Obrázek č.29: Zjednodušené blokové schéma bloku Adaptace

Blok *Adaptace* se skládá z části, která vytváří rozdíl mezi vstupními vzorky, bloku *Abs*, bloku *Multiport Switch* a části vytváření nové výšky schodu.

Princip tohoto je následující. Po příchodu výstupního impulsu modulátoru, který v bloku *Adaptace* tvoří vstupní impuls, se od něj odečte hodnota impulsu minulého. Tento rozdíl projde přes blok absolutní hodnoty *Abs* a stává se řídicím signálem pro blok *Multiport Switch*. Tento blok nastaví svůj výstup právě podle hodnoty na svém řídicím vstupu. Jak pracuje blok *Multiport Switch*, je patrné z tabulky č.2. Poslední částí bloku *Adaptace* je část, ve které se generuje nová hodnota výšky schodu. S každou periodou se tato zdvojnásobuje, pólí nebo za jistých podmínek může zůstat stejná (tedy násobená číslem 1). Tento stav je například na začátku simulace.

Tabulka č.3.1: Nastavení výstupní hodnoty bloku Multiport Switch

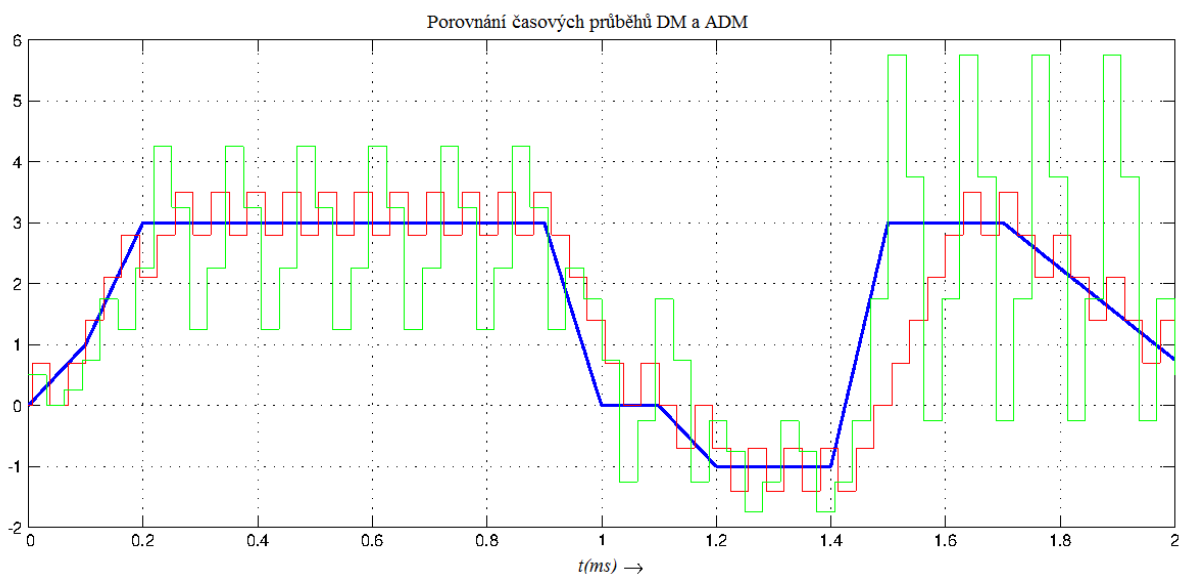
Řídicí signál	Význam řídicího signálu	Výstupní hodnota
0	dva po sobě jdoucí impulsy vstupního signálu jsou stejné	2
2	dva po sobě jdoucí impulsy vstupního signálu jsou různé	0,5
Jakákoliv jiná hodnota	Například začátek simulace	1



Obrázek č.30: Zjednodušené blokové schéma bloku Integrátor

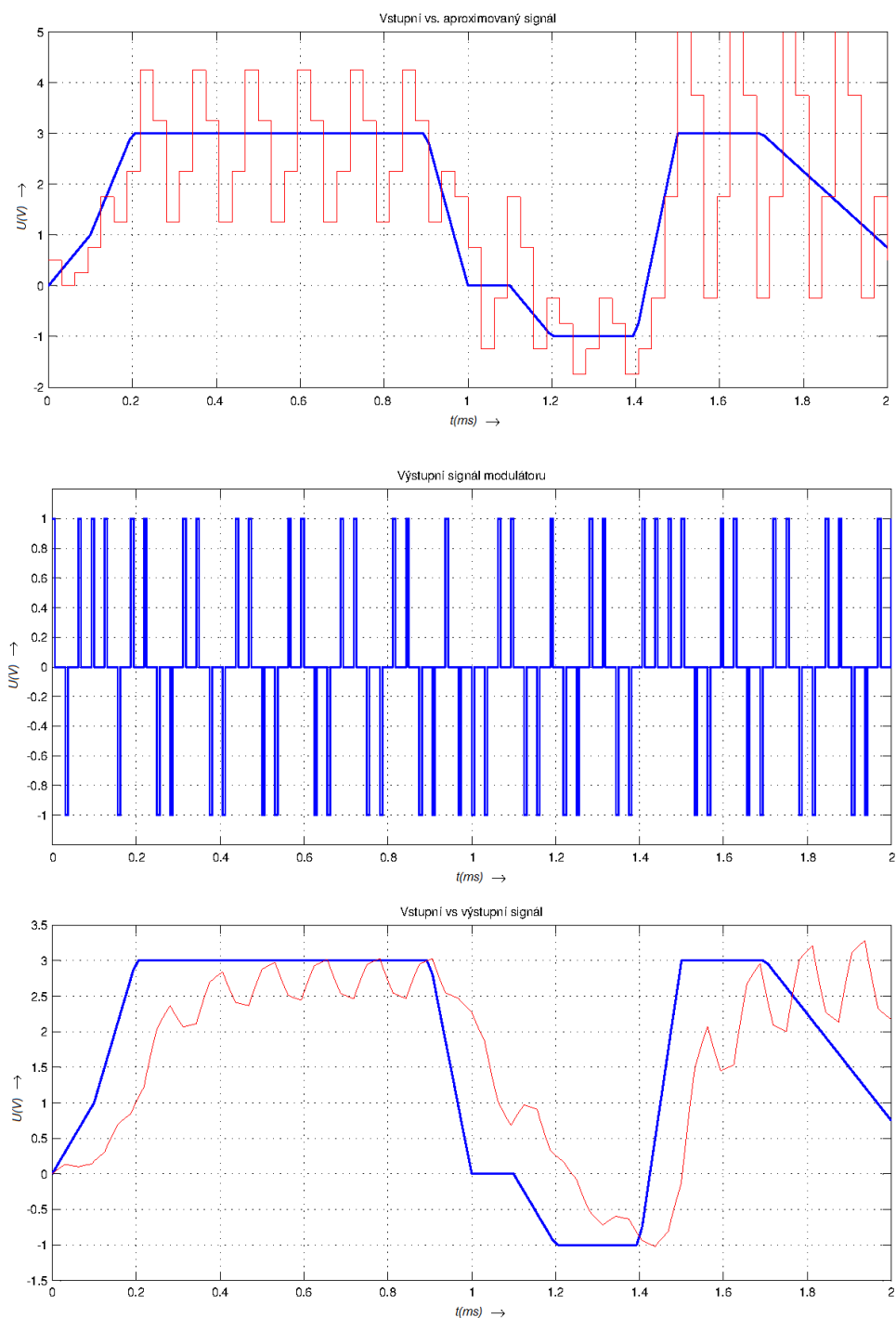
Původní blok *Integrátor* se skládal pouze z bloku diskrétního integrátoru, u něhož je statická hodnota výšky schodu zakomponována formou hodnoty *Gain value*. V případě ADM je tedy hodnota *Gain value* nastavena na 1 a výška schodu aproximovaného signálu je měněna pomocí násobení integrovaného signálu a hodnoty na druhém vstupu do bloku *Integrator*.

Na obrázku č.31 je porovnání aproximačních signálů při použití DM a ADM s Winklerovým algoritmem.



Obrázek č.31 Porovnání časových průběhů aproximačních signálů DM a ADM

Na tomto obrázku jsou znázorněny tři časové průběhy. Modrou barvou je vyznačen vstupní signál, červenou barvou je vyznačen aproximační signál DM a zelenou barvou je vyznačen aproximační signál ADM. Jak je na tomto obrázku vidět, při použití DM dochází k přetížení kodéru, ovšem při použití ADM k přetížení kodéru nedochází.

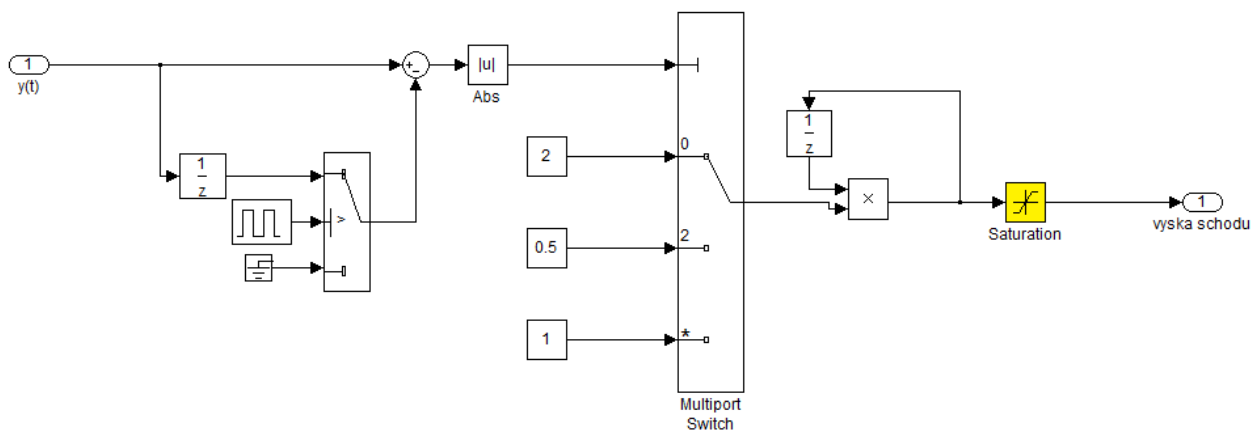


Obrázek č.32: Časové průběhy ADM bez omezení výšky schodu

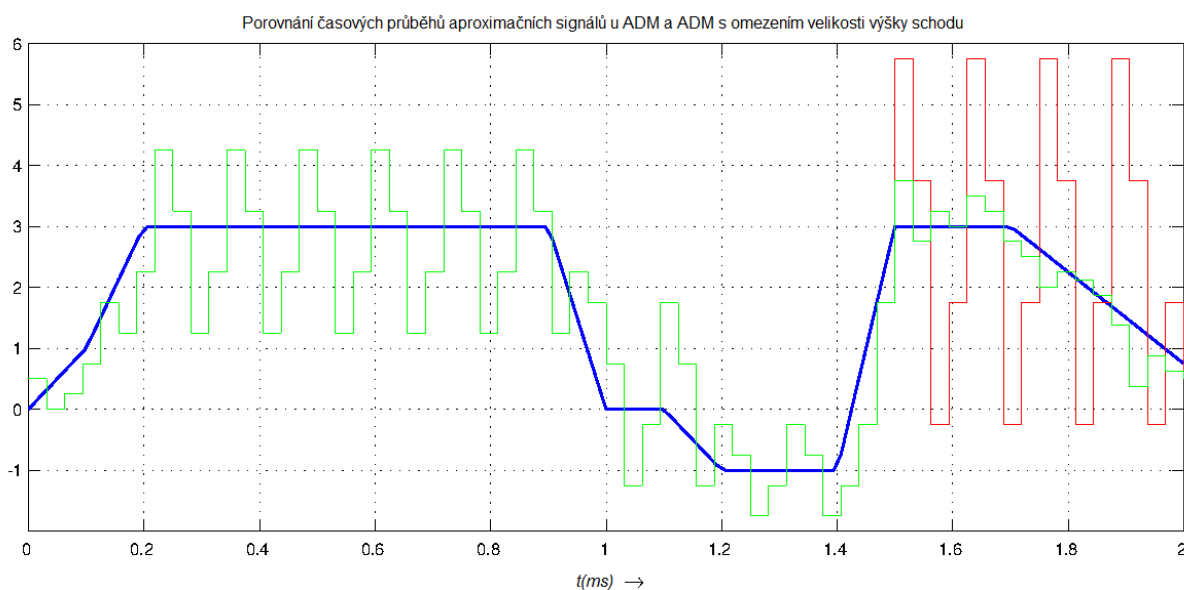
Jak je možno odpozorovat z obrázku č.31, může dojít při použití Winklerova algoritmu k situaci, kdy se velikost schodu hodně zvyšuje a velmi málo snižuje. Vlivem toho se může stát, že časem bude výška schodu tak velká, že nebude možno přenesený signál korektně demodulovat. Tato situace je znázorněna na obrázku č.32, kde od času cca 1,8ms přestává výstupní signál sledovat vstupní signál a dochází tak ke značnému zkreslení.

Tento problém se řeší nastavením horního a dolního limitu pro velikost schodu. Vhodnou komponentou v prostředí Matlab-Simulink je blok *Saturation*, v němž je možno nastavit oba tyto limity.

Blokové schéma bloku *Adaptace* s omezovačem *Saturation* je zobrazeno na následujícím obrázku.

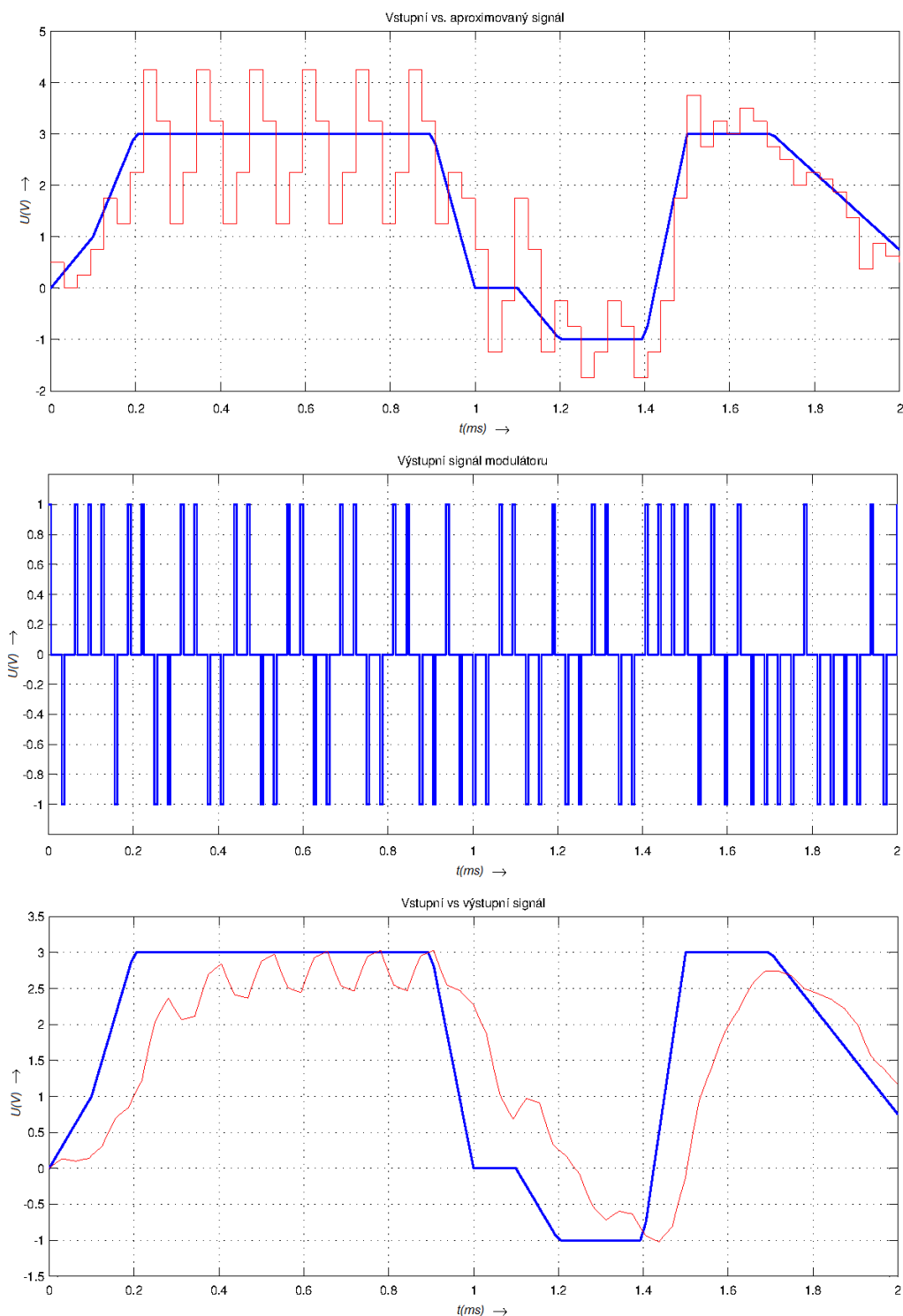


Obrázek č.33: Zjednodušené blokové schéma bloku Adaptace obsahující blok Saturation



Obrázek č.34: Porovnání časových průběhů aproximačních signálů

Na obrázku č. 34 je zobrazeno porovnání časových průběhů aproximačních signálů ADM a ADM s omezením výšky schodu. Modrou barvou je naznačen vstupní signál, červenou barvou aproximační signál ADM bez omezení výšky schodu a zelenou barvou je naznačen aproximační signál ADM s omezenou výškou schodu. Výška schodu byla omezena na rozsah hodnot 0,125 až 2. Jelikož se výška schodu začala zvyšovat nad hodnotu 2 až v čase cca 1,5ms a pod hodnotu 0,125 neklesla ani jednou, jsou až do této doby oba aproximační signály totožné.



Obrázek č.35: Časové průběhy ADM s omezením výšky schodu

Na obrázku č.35 jsou zobrazeny časové průběhy ADM s omezením výšky schodu. Při porovnání třetích časových průběhů z obrázků č.32 a č.35, je zřejmé, že při omezení výšky schodu sleduje výstupní signál vstupní lépe, nedochází tedy k takovému zkreslení jako v případě použití ADM bez omezení výšky schodu.

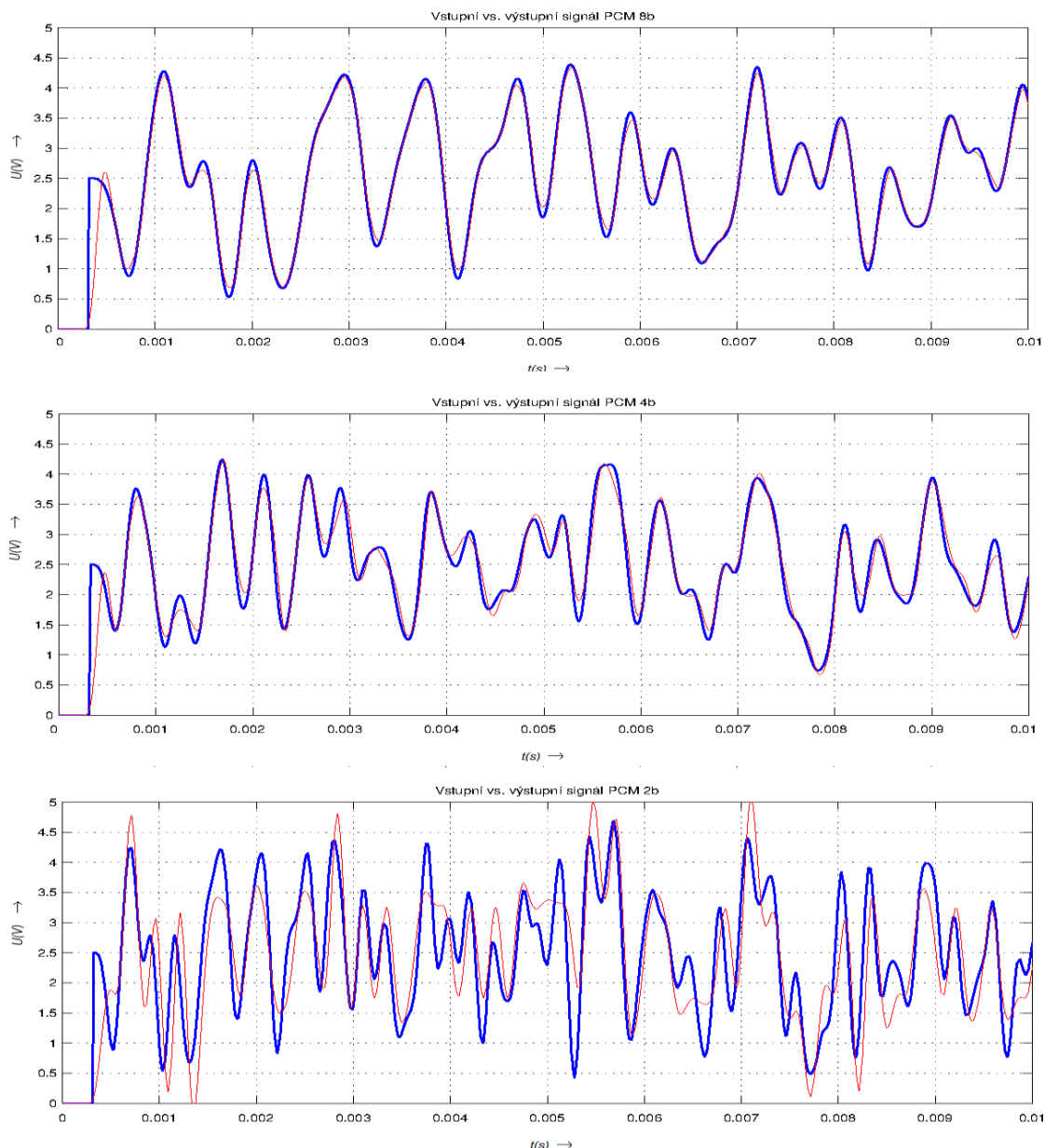
Při použití adaptabilní delta modulace nastává ještě jeden problém. Zatím co u klasické modulace delta, znal demodulátor výšku schodu, která byla po celou dobu stejná, u adaptabilní modulace delta, je nutné demodulátoru výšku schodu nějakým způsobem stále sdělovat.

4. Porovnání časových průběhů signálů získaných simulacemi

V této kapitole se práce zaměřuje na porovnání časových průběhů jednotlivých impulzových modulací a zhodnocení jejich závislosti na parametrech vstupního signálu a vzorkovací frekvence.

4.1 Porovnání časových průběhů PCM

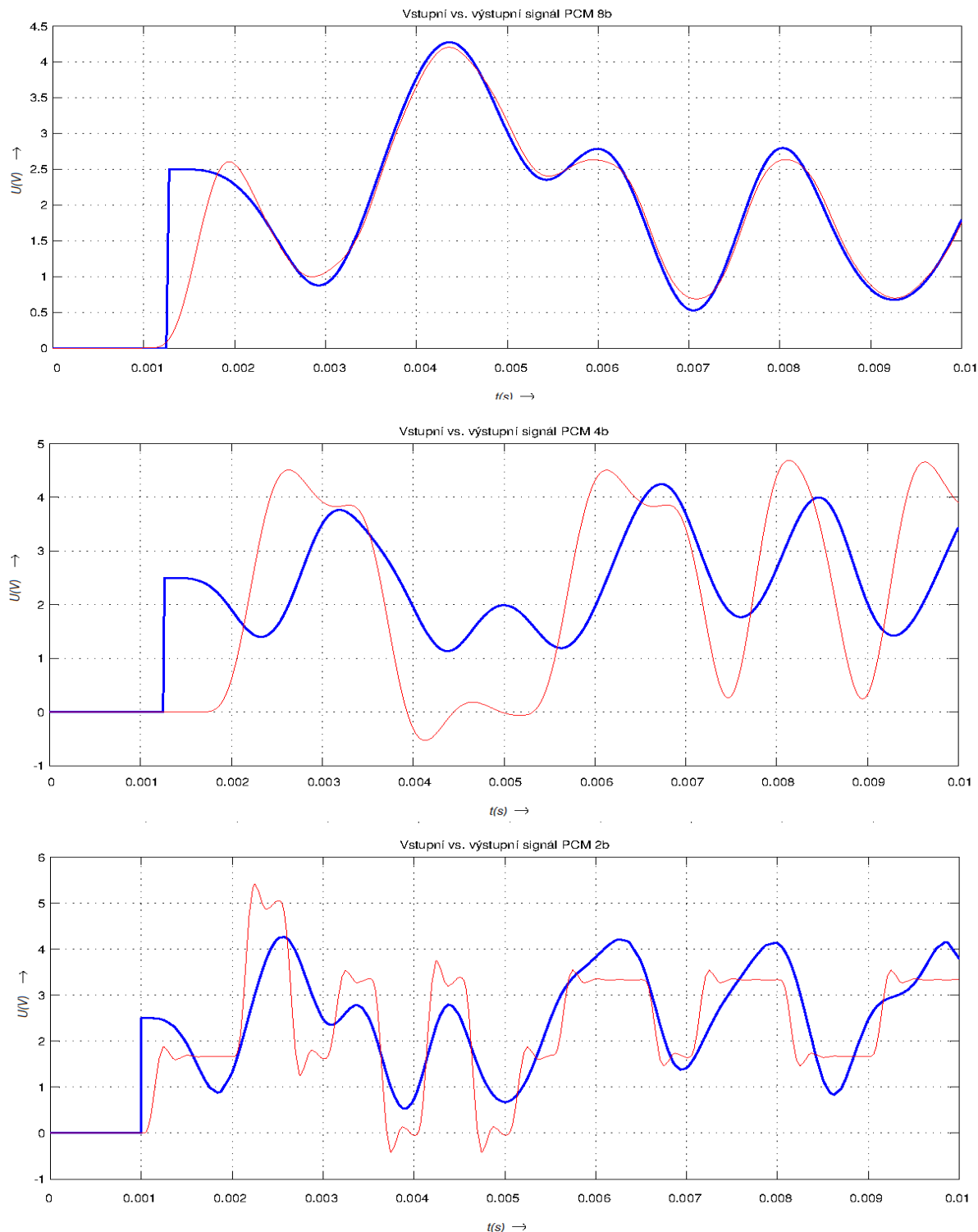
Na Obrázku č.36 jsou zachyceny časové průběhy vstupního a výstupního signálu při modulaci PCM pro 8bitovou, 4bitovou a 2bitovou modulaci při vzorkovací frekvenci 8kHz. Modrou barvou je znázorněn vstupní signál, červenou barvou je znázorněn výstupní signál.



Obrázek č.36: Časové průběhy vstupního a výstupního signálu PCM

Jak je z tohoto obrázku patrné, při použití 8bitové PCM je rozdíl mezi vstupním a výstupním signálem minimální, při použití 4bitové PCM se začínají projevovat malé rozdíly a při použití 2bitové PCM jsou již tyto rozdíly velké. Tyto rozdíly jsou dány kvantizací. I přes zobrazené rozdíly, je možno konstatovat, že všechny 3 případy modulace PCM jsou přijatelné.

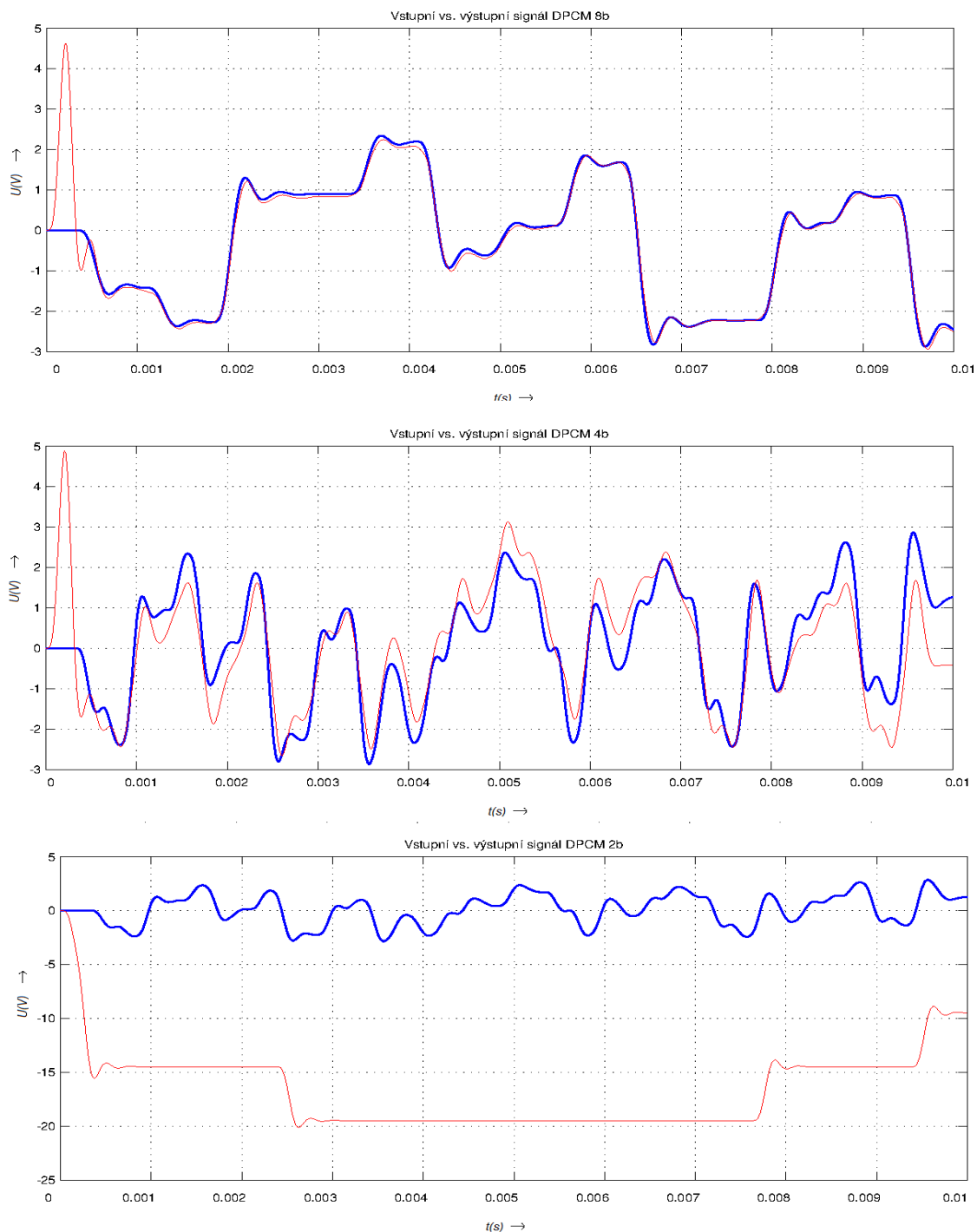
Na Obrázku č.37 jsou zachyceny časové průběhy vstupního a výstupního signálu při modulaci PCM pro 8bitovou, 4bitovou a 2bitovou modulaci nyní však se vzorkovací frekvencí 8kHz. Modrou barvou je znázorněn vstupní signál, červenou barvou je znázorněn výstupní signál.



Obrázek č.37: Časové průběhy vstupního a výstupního signálu PCM

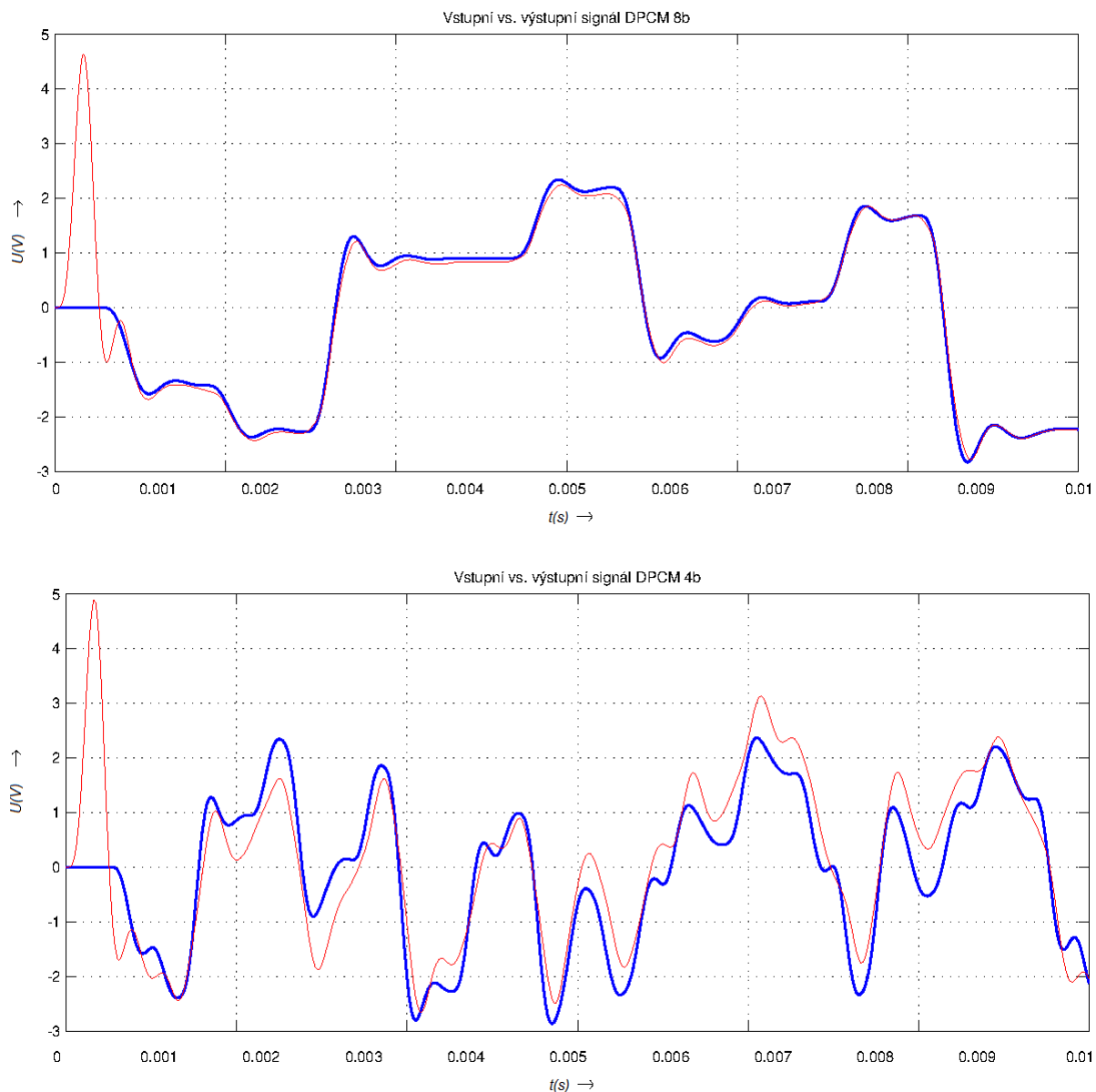
Jak je patrné při porovnání obrázků č.36 a č.37, PCM je jen velmi málo závislá na velikosti vzorkovací frekvence. Při použití vzorkovací frekvence 8kHz (obrázek č.36), výstupní signál relativně pěkně sleduje vstupní signál. V případě použití vzorkovací frekvence 2kHz, již vznikají větší rozdíly. I přes ně, lze považovat tuto modulaci za vhodnou.

4.2 Porovnání časových průběhů DPCM



Obrázek č.38: Časové průběhy vstupního a výstupního signálu DPCM

Na obrázku č. 38 jsou znázorněny časové průběhy vstupního a výstupního signálu DPCM pro 8bitovou, 4bitovou a 2bitovou modulaci se vzorkovací frekvencí 8kHz. Jak je z těchto průběhů patrné, 8bitová i 4bitová DPCM má téměř stejné výsledky jako 8bitová a 4bitová PCM, jediným velkým rozdílem je skok na začátku simulace, který je dán nastavením počátečních podmínek v modulátoru a demodulátoru. Co se týče 2bitové DPCM, výstupní signál téměř vůbec nesleduje vstupní signál. Tento problém je dán návrhem a konstrukcí modulátoru.

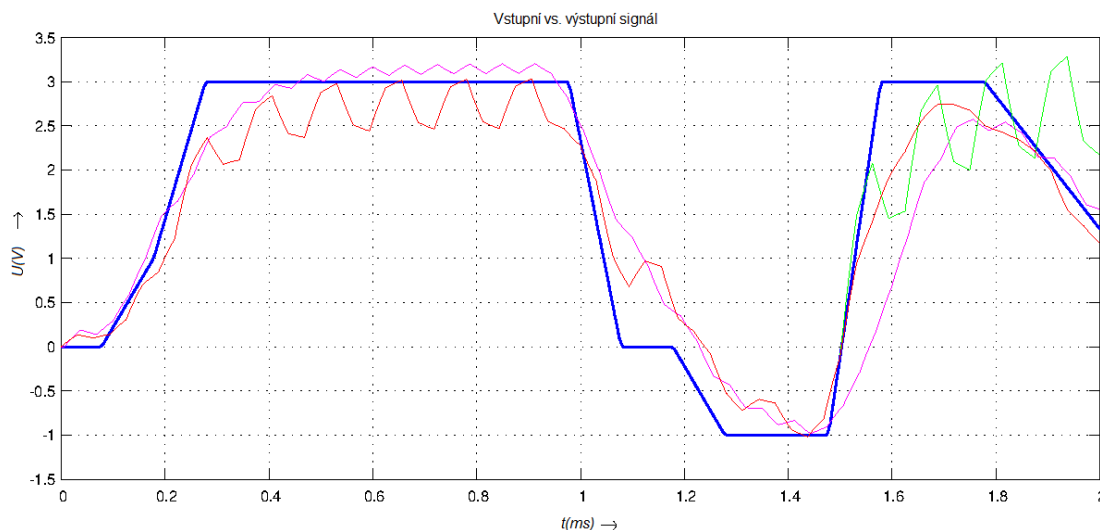


Obrázek č.39: Časové průběhy vstupního a výstupního signálu DPCM

Na obrázku č. 39 jsou znázorněny časové průběhy vstupního a výstupního signálu DPCM pro 8bitovou a 4bitovou modulaci se vzorkovací frekvencí 2kHz. 2bitová modulace zde již není zobrazena, protože již při vzorkovací frekvenci 8kHz nebyl výstupní průběh dobrý. Stejně jako tomu bylo u PCM, také DPCM není nijak výrazně závislá na hodnotě vzorkovací frekvence.

4.3 Porovnání časových průběhů DM a ADM

Tato část kapitoly se věnuje porovnání výstupních signálů DM, ADM bez omezení výšky schodu a ADM s omezením výšky schodu. Jak již bylo naznačeno v odstavcích týkajících se ADM, není-li použito omezení výšky schodu, dojde k rozkmitání tohoto parametru do velkých hodnot, díky čemuž dochází ke zkreslení danému odvrácením výstupního signálu od vstupního signálu. Na obrázku č.40 jsou zobrazeny celkem 4 časové průběhy. Modrou barvou je znázorněn vstupní signál, fialovou barvou výstupní signál při použití DM, zelenou barvou výstupní signál při použití ADM bez omezení výšky schodu a červenou barvou výstupní signál při použití ADM s omezením výšky schodu.



Obrázek č.40: Časové průběhy vstupního a výstupního signálu

Z obrázku je patrné, že při pozvolné změně vstupního signálu, mají všechny 3 modulační metody téměř stejné výsledky. Rozdíl však přichází v okamžiku, kdy se ve vstupním signálu objeví prudší změna. Při této situaci dosahují lepších výsledků oba typy ADM. Při následném ustálení signálu však výstupní signál ADM, u které není zavedeno omezování výšky schodu, začíná kmitat a přestává tak sledovat průběh vstupního napětí. Při použití ADM s omezením výšky schodu, k tomuto jevu nedochází.

Při porovnání všech těchto tří možností z pohledu jak výstupní signál sleduje vstupní signál, se jeví jako ta s nejlepšími výsledky právě ADM s omezením výšky schodu.

5. Programování impulzových modulací

V této kapitole se práce zaměřuje na popis návrhu programů, realizujících impulzové modulace v mikrokontroléru ATmega16.

Celé programování aplikací impulzových modulací do mikrokontroléru ATmega16 probíhalo ve vývojovém prostředí AVR Studio 4. Na výběr bylo programování v jazyku C, nebo programování v Assembleru. Z hlediska velkého snížení obtížnosti byla zvolena první možnost, tedy jazyk C.

5.1 Programování modulace PCM

První impulzovou modulací, která byla v rámci této diplomové práce programována, je modulace PCM. Programování všech modulací vycházelo z velké míry z modelů vytvořených ve vývojovém prostředí Matlab-Simulink.

Podle teorie popsané v kapitole 3, je prvním krokem pulzně kódové modulace operace **vzorkování**. Díky integrovanému A/D převodníku v používaném mikrokontroléru byl tento krok značně zjednodušen. Jediné, co bylo potřeba provést, je správné nastavení parametrů A/D převodníku. K těmto parametrům patří zejména:

úroveň referenčního napětí
povolení A/D převodu

Tyto parametry jsou v aplikaci nastaveny pomocí funkce *ADC_Init*. Jelikož pro danou aplikaci bylo požadované referenční napětí $U_{REF} = 5V$, přivedených na referenční vstup mikrokontroléru ze zdroje, nebylo potřeba nijak měnit hodnoty v registru *ADMUX*, který má na starosti, mimo jiné, také volbu referenčního napětí pro A/D převodník. V následující ukázce je zobrazen kód, který by znamenal volbu interního referenčního napětí $U_{REF}=2,56V$. [9]

```
ADMUX |= (1<<REFS1) | (1<<REFS0); // Nastavení interního referenčního  
                                     // napětí 2.56V
```

Sebelepší nastavení A/D převodníku by bylo k ničemu, kdyby samotný A/D převod nebyl povolen. To se uskuteční nastavením povolovacího bitu *ADEN* v registru *ADCSRA* na hodnotu logické jedničky. [9]

```
ADCSRA |= (1<<ADEN); // Povolení A/D převodu
```

Druhou funkcí, která se stará o bezproblémové provedení A/D převodu, je funkce *Read_ADC*. Předchozí funkce zajišťovala pouze inicializaci A/D převodníku, tedy jeho správné nastavení, kdežto tahle funkce vyčítá hodnoty z A/D převodníku. Může přijít otázka, proč neprovádět obě operace v jedné funkci. To by samozřejmě bylo možné, avšak každá inicializace A/D převodníku, stejně jako samotné vyčítání hodnoty trvá jistou dobu, proto v rámci úspory času, je vhodnější provést inicializaci pouze jednou na začátku aplikace a dále již pouze vyčítat hodnoty.

Kód funkce *Read_ADC* je naznačen níže.

```

uint16_t Read_ADC(uint8_t channel)
{
    ADMUX |= channel & 0x07;           //Výběr kanálu
    ADCSRA |= (1<<ADSC);              //Zapnutí A/D převodu
    while(!(ADCSRA & (1<<ADIF))); /*Čekání na konec A/D převodu, ten je
    signalizován nastavením bitu ADIF na hodnotu 1*/
    return(ADC);                      //Funkce vrací hodnotu vstupního signálu
}

```

Jak je z tohoto kódu patrné, jako první se musí provést výběr kanálu. V této práci byl používán téměř výhradně kanál 0. Po výběru kanálu již lze provést samotný A/D převod, který je zapínán nastavením bitu ADSC v registru ADCSRA, tedy ve stejném registru, ve kterém se ve funkci *ADC_Init* nastavoval povolovací bit ADEN. Každý A/D převod trvá určitou dobu, roto je nutné počkat na jeho ukončení, které je signalizované nastavením bitu ADIF na logickou hodnotu 1. Tento bit je podobně jako bity ADEN a ADSC umístěn v registru ADCSRA. Jakmile je tedy A/D převod ukončen, vrátí funkce *Read_ADC* hodnotu odpovídající úrovni vstupního signálu.

Druhou operací v impulzové modulaci PCM je **kvantování**. Opět díky integrovanému A/D převodníku je tato operace stejně jako minulá již hotová. Protože mikrokontrolér ATmega16 disponuje 10bitovým A/D převodníkem, je nutno hodnotu od něj získanou převést na hodnotu odpovídající počtu bitů, požadované modulace, tedy provést tzv. překvantování .

Maximální číslo získané díky 10bitovému A/D převodu je podle vztahu č.3 1023. Je tedy možné dosáhnout 1024 úrovní. Pomocí 8bitového A/D převodu lze dosáhnout 256 úrovní atd. Převod 10bitového čísla na číslo 8bitové se tedy dá uskutečnit jeho dělením číslem 4. V tabulce č.5.1, jsou uvedeny dělicí poměry pro převod 10bitového čísla na čísla s nbity.

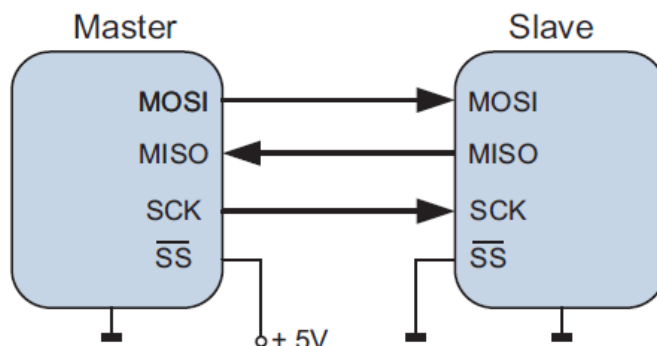
Tabulka č.5.1: Dělicí poměry pro převod z 10bitového čísla

n bitů	Počet úrovní	Dělicí poměr
10	1024	-
8	256	4
4	16	64
2	4	256

Posledním krokem PCM modulace je **kódování**. Pro tuto operaci byly v práci vytvořeny dva návrhové vzory. První z nich využívá ke kódování a následnému přenosu funkci mikrokontroléru SPI. Druhý vzor této funkce mikrokontroléru nevyužívá, a operaci kódování je u něj prováděna jako podíl vstupní hodnoty a jistých čísel, jakých bude popsáno dále v textu, a jeho okamžité vysílání k demodulátoru.

Návrhový vzor využívající SPI

SPI (Serial Peripheral Interface) je komunikační rozhraní mezi dvěma procesory využívající synchronizovanou sériovou linku. Celé rozhraní se skládá ze tří vodičů. Na obrázku č.42, je znázorněno propojení obou zařízení. [9]



Obrázek č.41: Propojení dvou zařízení pomocí SPI [9]

Jedno zařízení tzv. MASTER může vysílat data po lince MOSI (Master Output Slave Input) a přijímat data po lince MISO (Master Input Slave Output). Jak bylo zmíněno výše, jedná se o synchronní rozhraní, je tedy nutné aby jedno ze zařízení spolu komunikujících vysílal synchronizační signál. Tím je vždy zařízení MASTER. Zařízení vykonávající funkci SLAVE tento hodinový signál pouze přijímá. Volba, zda bude dané zařízení MASTER nebo SLAVE se provádí nastavením bitu $\overline{SS}$ na logickou hodnotu 1 nebo 0, tak jak je zobrazeno na obrázku.[9]

Pro využití funkce SPI v modulaci PCM je nutné nejprve nastavit modulátor i demodulátor. K tomu slouží čtyři jednoduché funkce. Dvě z nich jsou určeny pro modulátor, který v této aplikaci bude vystupovat jako MASTER a zbylé dvě jsou určeny pro demodulátor, který vystupuje v roli SLAVE. Všechny tyto funkce jsou zobrazeny a popsány níže.

První z těchto funkcí je *InitMasterSPI* [9]

```
void InitMasterSPI(void)                /*inicializační funkce pro SPI MASTER*/
{
    DDRB = (1<<PB4) | (1<<PB5) | (1<<PB7);
    /*zařízení se nastaví jako MASTER, na pinu B5 je MOSI a na pinu B7 je
    hodinový signál*/

    SPCR = ((1<<SPE) | (1<<MSTR) | (1<<SPR1) | (1<<SPR0));
    /*povolení SPI, nastavení MASTER, dělení frekvence /128 */
}
```

Funkce *InitMasterSPI* nastavuje zařízení do režimu MASTER, to se provede přivedením logické jedničky na pin PB4, reprezentující bit $\overline{SS}$. Dále nastaví pin PB5 na funkci MOSI a pin PB7 na funkci SCK. Všechna tato nastavení se prováděla v registru DDR (Data Direction Register) portu B mikrokontroléru, je však nutné provést jistá nastavení v registru SPCR (SPI Control Register). [8]

Těmito nastaveními jsou:

povolení funkce SPI – nastavení bitu SPE na hodnotu logické jedničky,
nastavení zařízení jako MASTER – nastavení bitu MSTR na hodnotu logické jedničky,
a dělení frekvence – bity SPR1 a SPR0.

Další z těchto funkcí je *SPI_MasterTransmit* [9]

```
void SPI_MasterTransmit(int byte)           //funkce pro vyslání dat po SPI
{
    SPDR = byte;
    /*do datového registru SPDR se vloží data, která se budou vysílat*/
    while(!(SPSR & (1<<SPIF)));           //čekání na dokončení přenosu
}
```

Funkce *SPI_MasterTransmit* zajišťuje vysílání dat od modulátoru k demodulátoru. Do datového registru SPDR se vloží data, která se budou vysílat. Tohle vysílání trvá určitou dobu, která je dána frekvencí oscilátoru a předděličkou frekvence.

Doba přenosu jednoho bytu je dána vztahem [9]

$$T = \frac{8 \cdot \text{prescaller}}{f_{osc}} \quad (9)$$

kde T(s) je doba přenosu jednoho bytu pomocí SPI,
prescaller (-) je hodnota předděličky frekvence oscilátoru a
f_{osc} (Hz) je hodnota frekvence oscilátoru.

Třetí z těchto funkcí, je *InitSlaveSPI* [9]

```
void InitSlaveSPI(void)                     //inicializační funkce pro SPI SLAVE
{
    SPCR = (1<<SPE);                       //povolení SPI
}
```

Funkce *InitSlaveSPI* zajišťuje povolení funkce SPI na straně demodulátoru, tedy u zařízení typu SLAVE. To se děje nastavením logické jedničky na místo bitu SPE v registru SPCR.

Poslední z těchto funkcí je *SPI_SlaveReceive* [9]

```
int SPI_SlaveReceive(void)
/*funkce pro načtení přijatých dat na straně SLAVE*/
{
    while(!(SPSR & (1<<SPIF)));           //čekání na přijetí všech dat
    return SPDR;                          //funkce vrací hodnotu datového registru
}
```

Funkce *SPI_SlaveReceive* ukládá přijímaná data do datového registru SPDR a po dokončení přenosu, který je signalizován nastavením bitu SPIF v registru SPSR na hodnotu logické jedničky, přijatá data vrátí do programu.

Modulátor

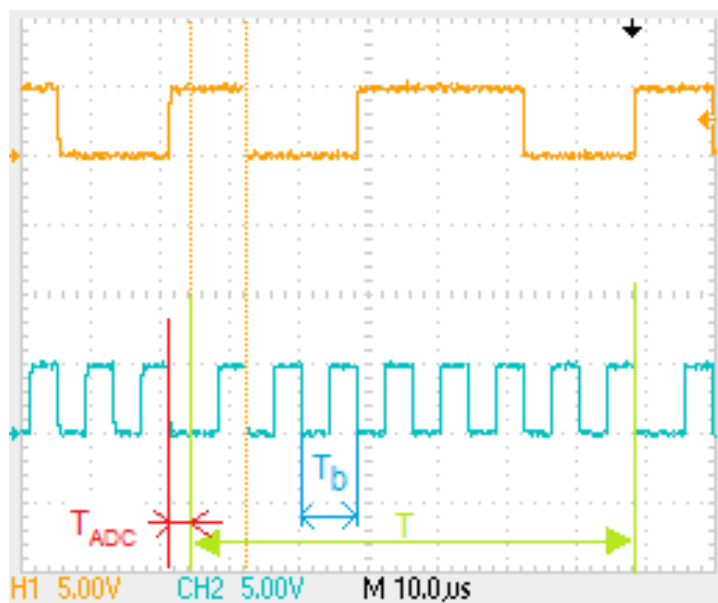
Na následujících řádcích je zobrazen a stručně popsán kód hlavního programu modulátoru PCM, který pro přenos využívá SPI rozhraní.

```
int main (void)                                //hlavní program
{
    InitMasterSPI();                            //volání inicializace SPI
    ADC_Init();                                //volání inicializace A/D převodníku

    while(1)                                    //nekonečná smyčka
    {
        Read_ADC(0);                            //ve které se načítají data z A/D
        SPI_MasterTransmit(ADC/4);              //a následně se posílají přes SPI
    }
    return 0;
}
```

Na začátku programu je inicializováno rozhraní SPI a A/D převodník. Poté se v nekonečné smyčce provádí vyčítání hodnot z A/D převodníku a jejich posílání pomocí rozhraní SPI. Protože je tato aplikace upravena pro 8bitovou PCM, je zde z důvodu přehlednosti pouze zapsán vstupní parametr funkce obsluhující odesílání přes SPI jako $ADC/4$. ADC je hodnota vrácená A/D převodníkem a 4 je koeficientem pro překvantování z 10bitové hodnoty na 8bitovou hodnotu. Viz tabulka č.5.1.

Na následujícím obrázku je zachycen průběh hodinového signálu a signálu posílaného přes SPI rozhraní.



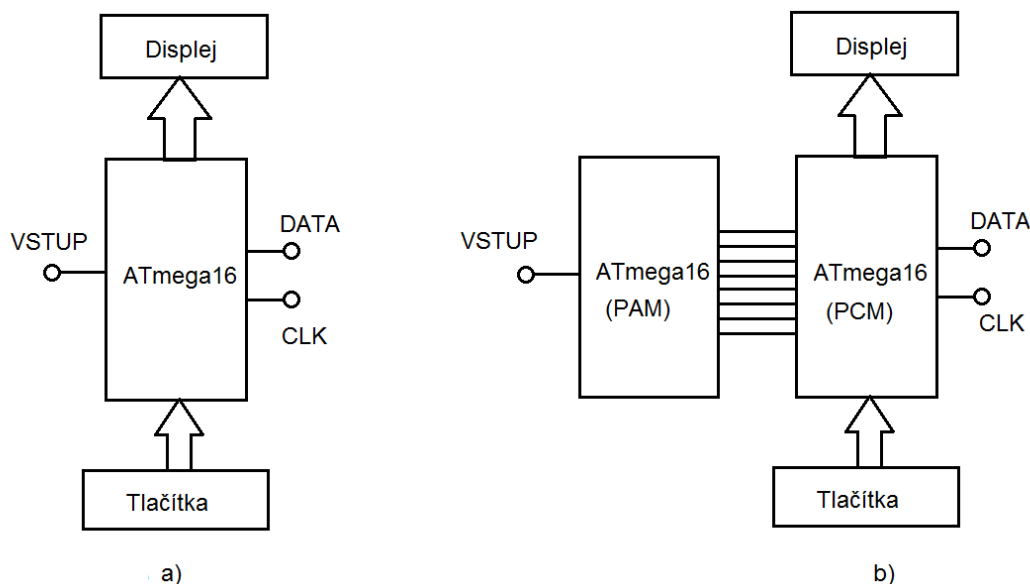
Obrázek č.42: Časové průběhy signálů při přenosu přes SPI rozhraní

Na obrázku č.42 je oranžovou barvou zachycen časový průběh výstupního signálu PCM a modrou barvou je zachycen časový průběh hodinového signálu. Dále je zde také naznačena perioda T zelenou barvou, a modře doba trvání jednoho bitu T_b . Zároveň je možné si na tomto průběhu všimnout časového zpoždění T_{ADC} , zobrazeno červeně. Tohle zpoždění odpovídá době A/D převodu.

Protože modulace PCM v sobě nemá definováno žádné takové zpoždění, je tedy nepřípustné, aby se ve výstupním signálu nacházelo. Řešením tohoto problému může být připojení externího A/D převodníku k mikrokontroléru, čímž odpadá nutnost provádět A/D převod přímo na mikrokontroléru a tím eliminovat dané zpoždění.

Návrhový vzor využívající SPI a externí A/D převodník

Jak bylo uvedeno v předchozím odstavci, z důvodu eliminace zpoždění daného A/D převodem, bylo dosavadní zapojení a program aplikace obměněno. Zapojení bylo obohaceno o druhý mikrokontrolér ATmega16, který zde však vykonává pouze převod A/D. V důsledku v podstatě realizuje modulaci PAM, u které je doba pulzu rovna době trvání vzorkovací periody.



Obrázek č.43: Zjednodušené blokové schéma modulátoru PCM

Na obrázku č.43 je zobrazeno zjednodušené blokové schéma PCM modulátoru, u kterého je prováděn A/D převod přímo na mikrokontroléru provádějící modulaci (a), a zjednodušené blokové schéma PCM modulátoru, u kterého je A/D převod realizován druhým mikrokontrolérem.

Obměněný program modulátoru PCM je zobrazen níže.

```
int main (void) //hlavní program
{
    InitMasterSPI(); //volání inicializace SPI
    DDRD = 0x00; //nastavení portu D na vstupní

    while(1) //nekonečná smyčka
    {
        ADCx = PIND; //do ADCx se uloží hodnota z portu D
        SPI_MasterTransmit(ADCx); //a následně se posílají přes SPI
    }
    return 0;
}
```

Výše zobrazený kód popisuje změněnou verzi programu realizujícího modulaci PCM. Hlavní změnou je odstranění funkce načítání hodnoty z A/D převodníku a nahrazení této funkce načítáním hodnoty ze vstupního portu D, na který je tato hodnota přivedena z druhého mikrokontroléru, realizujícího funkci A/D převodníku. Kód programu realizující A/D převod je popsán níže.

```
int main (void) //hlavní program
{
    ADC_Init(); //volání inicializace A/D převodníku
```

```

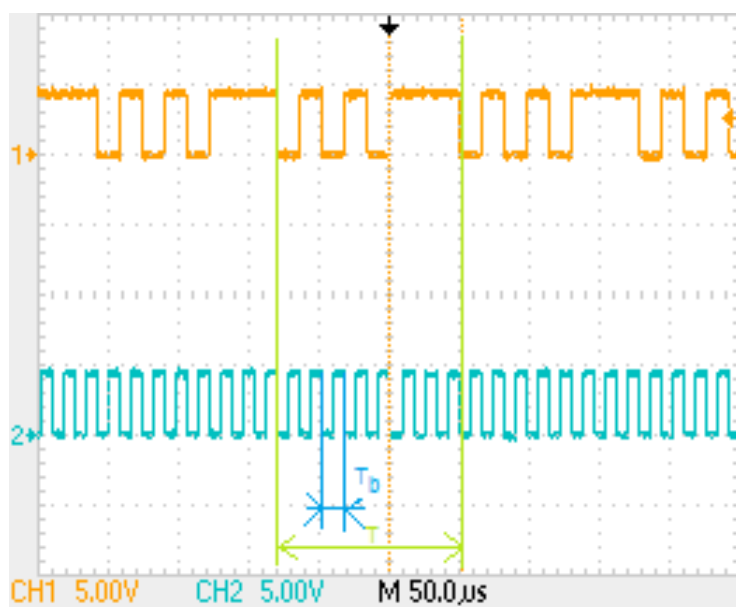
    DDRB = 0xff;           //nastavení portu B na výstupní

    while(1)               //nekonečná smyčka
    {
        Read_ADC(0);       //ve které se načítají data z A/D
        ADCx = ADC/4;      //jsou převedena z 10b na 8b hodnotu
        PORTB = ADCx;      //a jsou posílány na port B
    }
    return 0;
}

```

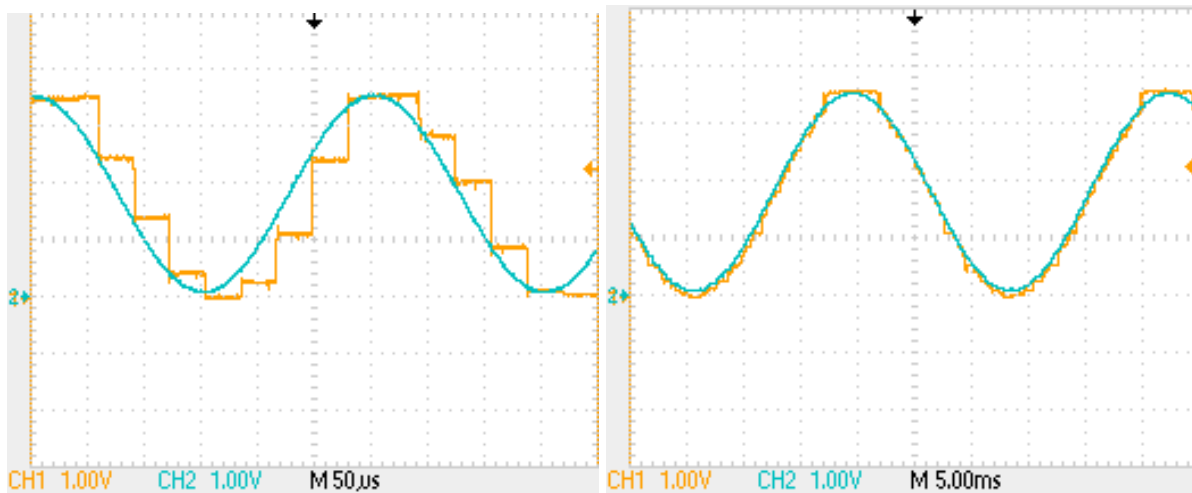
Program realizující A/D převod ve druhém mikrokontroléru inicializuje A/D převodník a nastaví port B jako výstupní. V nekonečné smyčce je poté prováděno načítání hodnot z A/D převodníku, upraví ji z 10bitové hodnoty na, v tomto případě, 8bitovou hodnotu, kterou poté pošle na port B. Výstupní port B tohoto mikrokontroléru je spojen se vstupním portem D mikrokontroléru realizujícího modulaci PCM.

Na následujícím obrázku jsou zobrazeny časové průběhy hodinového signálu a signálu posílaného přes SPI rozhraní.



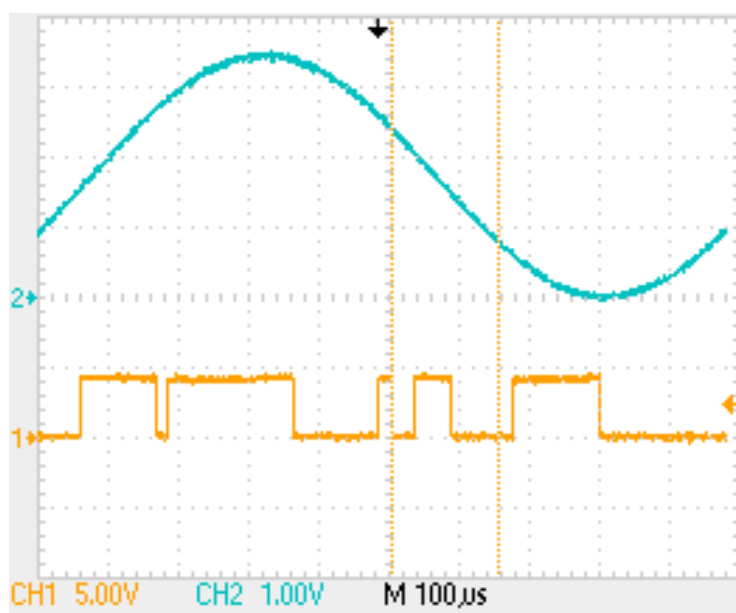
Obrázek č.44: Časové průběhy signálů při přenosu přes SPI rozhraní

Jak je z tohoto obrázku patrné, žádné zpoždění dané převodem A/D se zde nevyskytuje. Tímto je vyřešen problém vzniklý v předchozím případě, kdy jeden mikrokontrolér prováděl jak A/D převod, tak modulaci PCM.



Obrázek č.45: Časové průběhy vstupního signálu a výstupního signálu PAM v modulátoru PCM

Na obrázku č.45 jsou zachyceny časové průběhy vstupního signálu (modrý) a výstupního signálu PAM (oranžový). První z nich zachycuje časové průběhy při frekvenci vstupního signálu přibližně 3kHz, a druhý zachycuje tytéž průběhy, avšak při frekvenci vstupního signálu přibližně 30Hz.



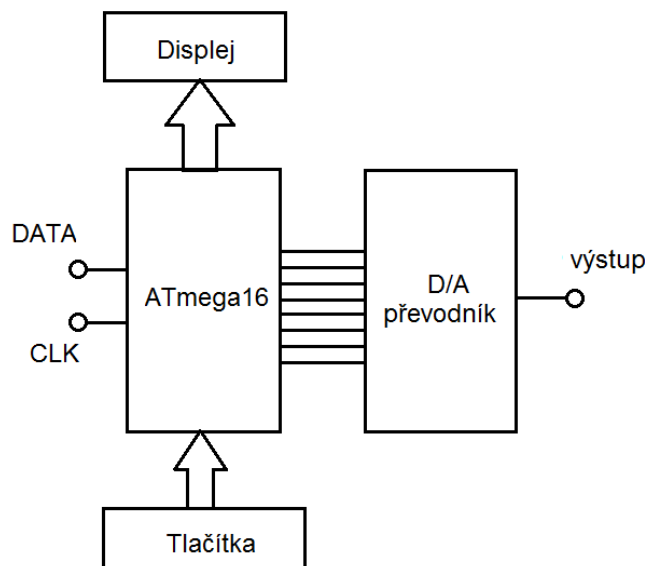
Obrázek č.46: Časové průběhy vstupního a výstupního signálu v modulátoru PCM

Na obrázku č.46 jsou zachyceny časové průběhy vstupního signálu o frekvenci přibližně 1kHz a výstupního signálu. Kurzory je vyznačena vzorkovací perioda modulátoru.

Demodulátor

Demodulátor pulzně kódové modulace plní, při použití tohoto návrhové vzoru, funkci zařízení SLAVE, podle definice tohoto rozhraní tedy přijímá data od modulátoru na pinu označeném MOSI viz. obrázek č.41.

Zjednodušené blokové schéma demodulátoru je znázorněno na následujícím obrázku.



Obrázek č.47: Zjednodušené blokové schéma demodulátoru

Jak je z obrázku č.47 patrné, demodulátor má dva vstupy a jeden výstup. Na jeden ze vstupů, označený DATA, je přiváděn výstupní signál z modulátoru, na druhý vstup, označený CLK, je přiváděn synchronizační signál z modulátoru. Protože mikrokontrolér ATmega16 nedisponuje D/A převodníkem, je nutné o něj demodulátor doplnit. Volba obvodu realizující D/A převod padla na integrovaný obvod DAC0832.

Hlavní část kódu aplikace provádějící demodulaci PCM je naznačen níže.

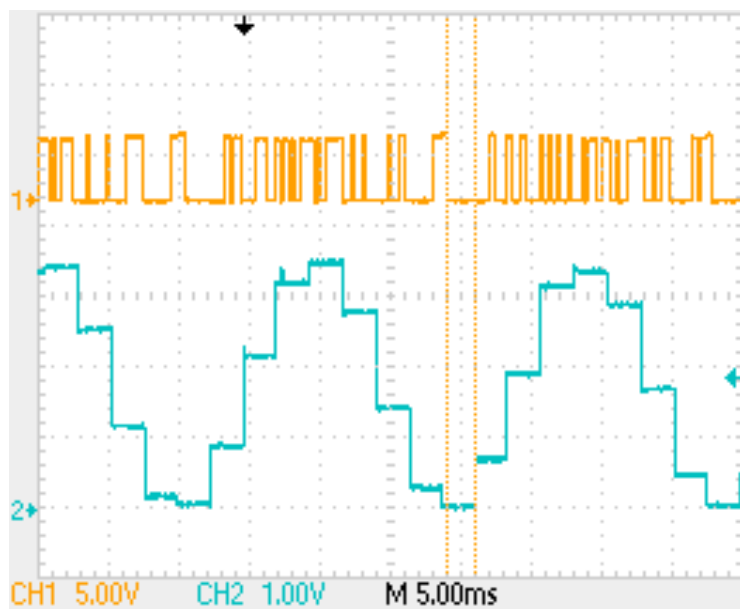
```
int main (void)
{
    InitSlaveSPI();           //inicializace SPI a nastavení režimu SLAVE
    DDRA = 0xff;              //nastavení portu A na výstupní

    while(1)
    {
        if((PINB & 0b10000000)!=0) //čekání na příchod 1.logické 1
        {
            while(1)           //začne se vykonávat nekonečná smyčka
            {
                SPI_SlaveReceive(); //ve které se přijmou data ze sběrnice SPI
                PORTA = SPDR;        //a následně se pošlou na výstupní port C
            }
        }
    }
    return 0;
}
```

Z předcházejícího kódu je patrné, že demodulátor nejprve provede inicializaci rozhraní SPI a nastavení typu zařízení SLAVE. V dalším kroku nastaví program port A jako výstupní. Tento port je připojen k D/A převodníku. Dále program čeká na příchod prvního synchronizačního

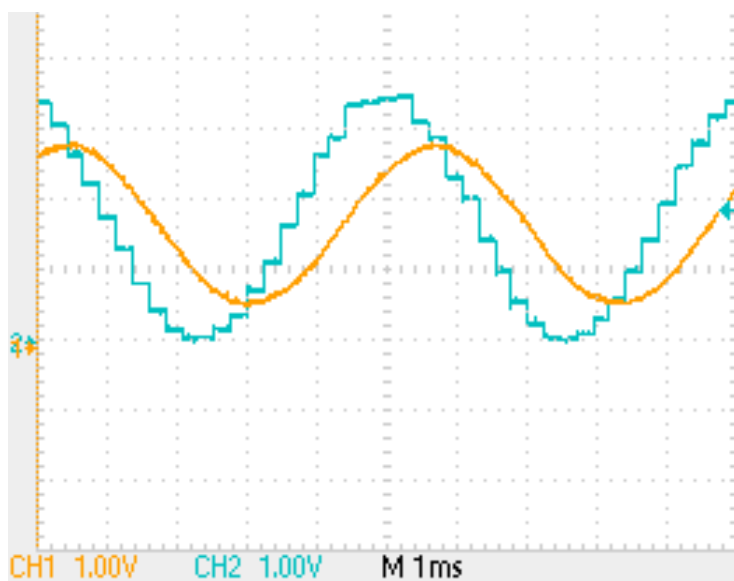
pulzu na vstupním pinu B7. Po jeho přijetí se začne vykonávat nekonečná smyčka programu, v níž je volána funkce `SPI_SlaveReceive`, vracející přijatou hodnotu z rozhraní SPI. Tato hodnota je posílána na výstupní port A. Dále s touto hodnotou pracuje obvod vykonávající funkci D/A převodníku.

Na následujících obrázcích jsou znázorněny časové průběhy signálů v demodulátoru.



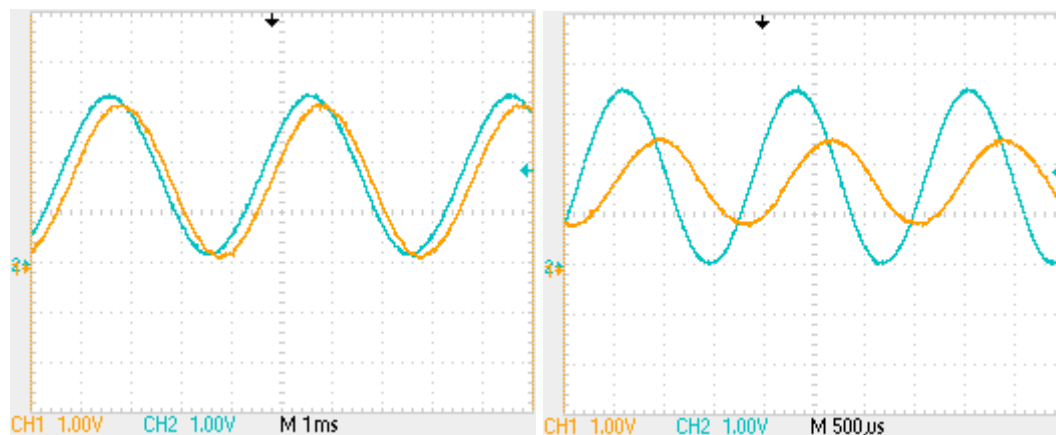
Obrázek č.48: Časové průběhy vstupního a výstupního signálu před filtrací v demodulátoru PCM

Na obrázku č.48 jsou zachyceny časové průběhy vstupního signálu PCM demodulátoru, který odpovídá vstupnímu signálu modulátoru PCM s frekvencí přibližně 500Hz. Tento průběh je zachycen oranžovou barvou. Druhý průběh reprezentuje výstupní signál PCM demodulátoru, který je však ještě pouze ve schodovitém tvaru. Tento signál odpovídá výstupnímu signálu PAM modulátoru PCM. Aby bylo možné tento signál považovat za výstupní signál demodulátoru, je nutné provést jeho vyhlazení rekonstrukčním filtrem.



Obrázek č.49: Časové průběhy výstupního signálu demodulátoru PCM před filtrací a po filtraci

Na obrázku č.49 jsou zachyceny časové průběhy výstupního signálu demodulátoru PCM, z nichž jeden (modrý) je před vyhlazením rekonstrukčním filtrem a druhý (oranžový), již tímto procesem prošel.



Obrázek č.50: Časové průběhy vstupního signálu modulátoru PCM a výstupního signálu demodulátoru PCM

Na obrázku č.50 jsou zachyceny časové průběhy vstupního signálu modulátoru (modře) a výstupního signálu demodulátoru (oranžově). Jak je možné z těchto průběhů vypočítat, nastává při přenosu signálu mírné zpoždění. První obrázek odpovídá frekvenci vstupního signálu 100Hz a druhý obrázek odpovídá frekvenci vstupního signálu 500Hz. Jak je vidět, s rostoucí frekvencí se časový průběh výstupního signálu demodulátoru více odchyluje od časového průběhu vstupního signálu modulátoru. Tato skutečnost je dána rekonstrukčním filtrem nízkého řádu, který má u vyšších frekvencí, i přesto, že tyto frekvence patří do propustného pásma tohoto filtru, větší útlum než u frekvencí nižších.

Návrhový vzor bez použití rozhraní SPI - modulátor

Na rozdíl od doposud popisovaného návrhového vzoru, který využíval rozhraní SPI, tento návrhový vzor rozhraní SPI nevyužívá. Místo toho využívá nastavování výstupního pinu mikrokontroléru na jednu ze dvou logických úrovní a okamžitého vysílání tohoto bitu směrem k demodulátoru. Ze zkušenosti nabyté při programování modulátoru využívajícího rozhraní SPI, bylo i v tomto návrhovém vzoru využito zapojení naznačené na obrázku č.44b, tedy zapojení využívající externího A/D převodníku.

Na níže zobrazeném kódu programu je vyobrazena hlavní funkce realizující samotnou modulaci.

```
void PCM (void)
{
    int hodnoty8[]={1, 2, 4, 8, 16, 32, 64, 128};
    /*definice dělicích hodnot pro 8bitovou PCM*/

    DDRD = 0x00;                //nastavení portu D na vstupní
    DDRB = 0xff;                //nastavení portu B na výstupní

    while(1)                    //nekonečná smyčka programu
    {
        PORTB |= (1<<7);        //nastavení výstupního pinu B7 do log. 1

        ADCx = PIND;            //načtení hodnoty z A/D převodníku

        for(int i = 7; i>=0 ;i--) //smyčka pro vykonání 8 cyklů
        {
            digital = ADCx / hodnoty8[i]; //výpočet výstupního bitu
            if(digital == 1)             //je-li výstupní bit 1
            {
                PORTB = 0b00100000;     //na pin B5 se pošle log. 1
                wait2();                 //doba vysílání výstupního bitu
                ADCx = ADCx-hodnoty8[i]; //přepočítání hodnoty ADCx
            }

            else                       //je-li výstupní bit 0
            {
                PORTB = 0b00000000;     //na pin B5 se pošle log. 0
                wait2();                 //doba vysílání výstupního bitu
                ADCx = ADCx-0;           //přepočítání hodnoty ADCx
            }
        }
    }
}
```

Jak je z tohoto kódu patrné, je složitější než ten, realizující modulaci s využitím SPI rozhraní. Pro lepší přehlednost je zde tento kód uveden pro 8bitovou modulaci. Jeho princip je následující: nejprve je potřeba si nadefinovat statické pole s hodnotami mocnin 2, které budou sloužit jako dělitel hodnoty získané z A/D převodníku, pro výpočet výstupního bitu signálu. Po definici vstupních a výstupních portů se program dostává do nekonečné smyčky, ve které se provádí samotný výpočet hodnoty výstupního bitu.

Pro lepší orientaci v časových průbězích je hodinový signál omezen pouze do formy vyslání krátkého pulzu na výstupním pinu B7 modulátoru, který označuje začátek přenosu n bitů. V tomto případě je impuls vyslán vždy po 8 bitech.

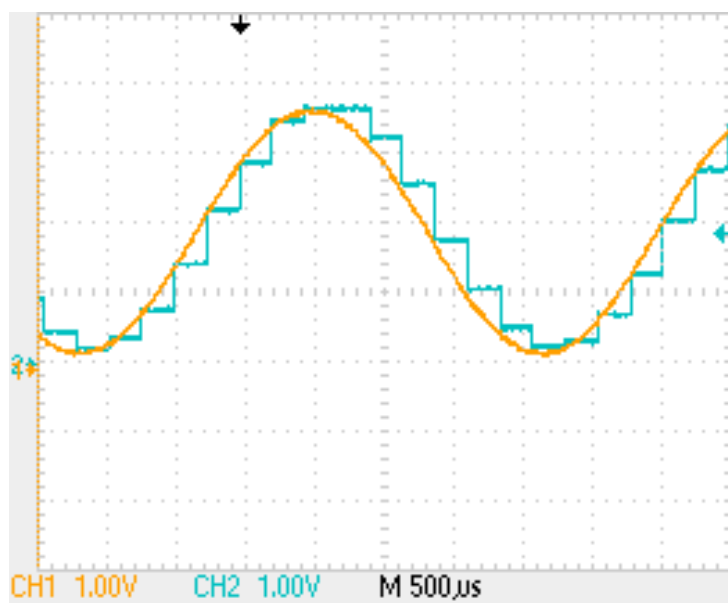
Dalším krokem programu je načtení hodnoty z A/D převodníku, které je realizováno stejně, jako u modulace využívající SPI rozhraní, tedy načtením hodnoty ze vstupního portu D.

V této fázi se program dostává do smyčky, v níž provede vysílání n bitů na výstupní pin B5.

V každé fázi této smyčky program provede výpočet hodnoty aktuálního bitu, který je výsledkem celočíselného dělení čísla $ADCx$ a příslušné hodnoty ze statického pole `hodnoty8`. Je-li výsledek tohoto dělení roven 1, vyšle se na výstupní pin B5 logická hodnota 1, v opačném případě se vyšle logická hodnota 0. Jelikož je potřeba dodržet dobu trvání výstupního bitu na hodnotě, odpovídající vzorkovací frekvenci, je nutné, aby každý tento vyslaný výstupní bit trval určitou dobu. O to se stará funkce `wait2()` realizující zpožďovací časovou smyčku.

Na závěr každého z n cyklů je potřeba změnit hodnotu $ADCx$ v závislosti na aktuální hodnotě výstupního bitu. Jak je naznačeno v kódu, je-li aktuální výstupní bit roven logické úrovni 1, musí se od hodnoty $ADCx$ odečíst příslušná hodnota z definovaného statického pole. V opačném případě zůstává hodnota $ADCx$ beze změny.

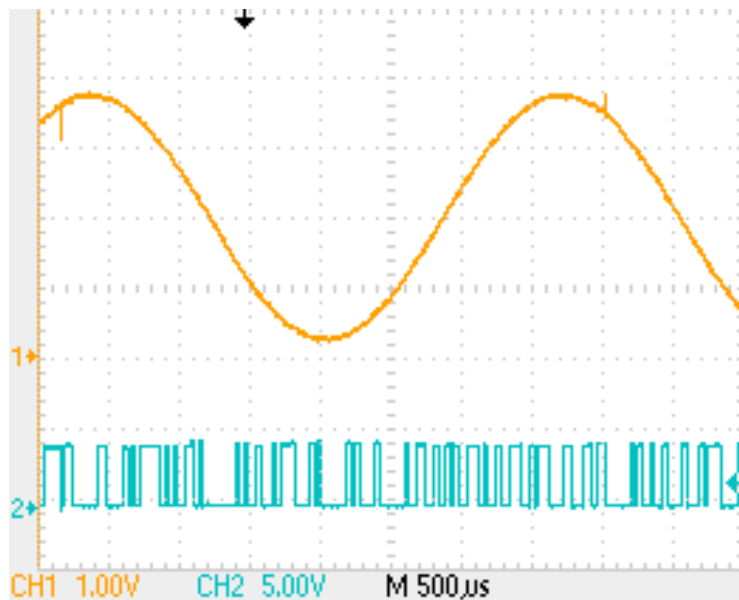
Na dalších obrázcích jsou znázorněny časové průběhy signálů v PCM modulátoru.



Obrázek č.51: Časové průběhy vstupního signálu a výstupního signálu PAM v modulátoru PCM

Na obrázku č.51 je zachycen časový průběh vstupního signálu (oranžový) o frekvenci přibližně 300Hz a výstupní signál PAM. Jak je z tohoto obrázku patrné, frekvence není nikterak vysoká, proto je z jedné periody vstupního signálu vzato hned několik vzorků.

Na obrázku č.52 jsou zachyceny časové průběhy vstupního a výstupního signálu v modulátoru PCM.



Obrázek č.52: Časové průběhy vstupního a výstupního signálu modulátoru PCM

Demodulátor modulace PCM bez využití rozhraní SPI

Demodulátor tohoto návrhové vzoru se od demodulátoru použitého v předcházejícím návrhovém vzoru mnoho neliší. Blokové schéma zapojení demodulátoru je naprosto stejné jako v předchozím případě, viz obrázek č.47. Zásadní rozdíl je v kódu aplikace, který je zobrazen níže. Pro větší přehlednost je uvažována 8bitová PCM modulace.

```
int main (void)
{
    int hodnoty8[8]={1, 2, 4, 8, 16, 32, 64, 128};
    /*definice násobících hodnot pro 8bitovou PCM*/

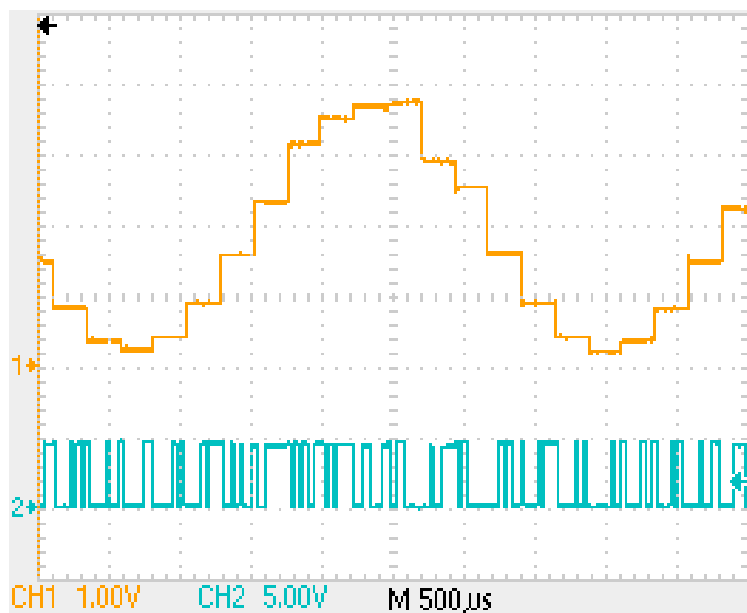
    DDRB = 0x00;                //nastavení portu B na vstupní
    DDRA = 0xff;                //nastavení portu A na výstupní

    while(1)
    {
        if((PINB & 0b10000000)!=0) //čekání na příchod 1.logické 1
        {
            while(1)            //začne se vykonávat nekonečná smyčka
            {
                vystup = 0;
                for(int i = 7; i>=0 ;i--)    //smyčka pro vykonání 8 cyklů
                {
                    vystup = vystup + PINB5 * hodnoty8[i]; //výpočet výst. hodnoty
                    wait2();    //doba mezi vysíláním jednotlivých bitů
                }
                PORTA = vystup;    //výst. Hodnota se nastaví na výst. port A
            }
        }
    }
    return 0;
}
```

Kód programu PCM demodulátoru začíná obdobně jako v případě PCM modulátoru definicí statického pole obsahujícího mocniny 2. Dále je nutné nastavit správně vstupní a výstupní porty. Stejně jako v případě demodulátoru, realizovaného podle návrhové vzoru využívajícího

rozhraní SPI, také demodulátor, realizovaný bez použití SPI rozhraní, čeká na příjem prvního synchronizačního pulzu na vstupním pinu B7. Po jeho příjmu se program dostává do nekonečné smyčky, v níž se nejprve výstupní hodnota nastaví na 0. Poté se v cyklu o 8 krocích provede přepočítání výstupní hodnoty podle zobrazeného vzorce. K aktuální výstupní hodnotě se přičte součin příslušné hodnoty ze statického pole `hodnoty8` a aktuálně přijímané hodnoty na vstupním pinu B5. Protože doba trvání aktuálního bitu je dána modulátorem, musí také demodulátor přijímat po stejnou dobu pouze jednu hodnotu. Doba příjmu jednoho bitu je dána obdobně jako u modulátoru doba vysílání jednoho bitu funkcí `wait2`. Po příjmu všech, v tomto případě 8, bitů, je cyklus ukončen a aktuální výstupní hodnota je nastavena na výstupní port A.

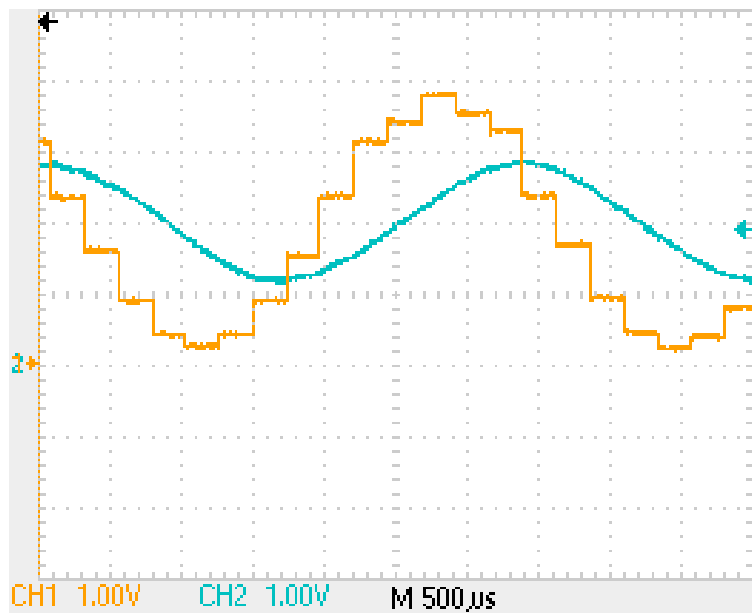
Na následujícím obrázku jsou znázorněny časové průběhy signálů v demodulátoru.



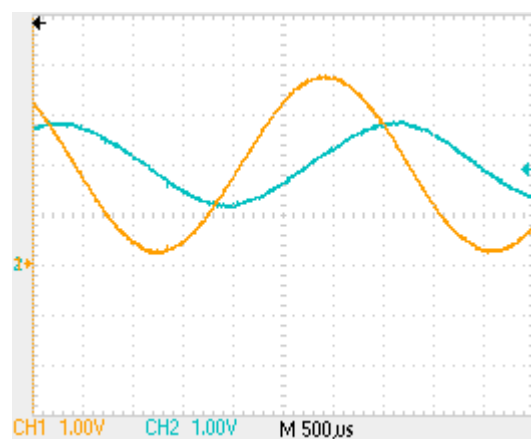
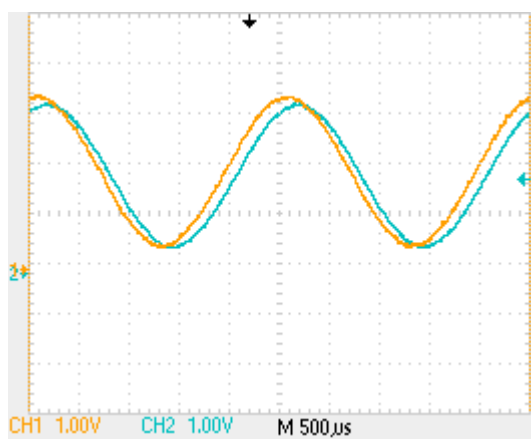
Obrázek č.53: Časové průběhy vstupního signálu a nefiltrovaného výstupního signálu v demodulátoru PCM

Na obrázku č.53 jsou zachyceny časové průběhy vstupního signálu demodulátoru PCM (modře) a výstupního signálu demodulátoru PCM ve schodovitém tvaru. Tento signál by měl odpovídat výstupnímu PAM signálu modulátoru. Aby bylo možné jej považovat za výstupní signál demodulátoru je nutné aby prošel rekonstrukčním filtrem.

Tato událost je zaznamenána na následujícím obrázku, kde je oranžovou barvou znázorněn signál před rekonstrukčním filtrem a modrou barvou signál po průchodu tímto filtrem.



Obrázek č.54: Časové průběhy výstupního signálu demodulátoru PCM před a po filtraci

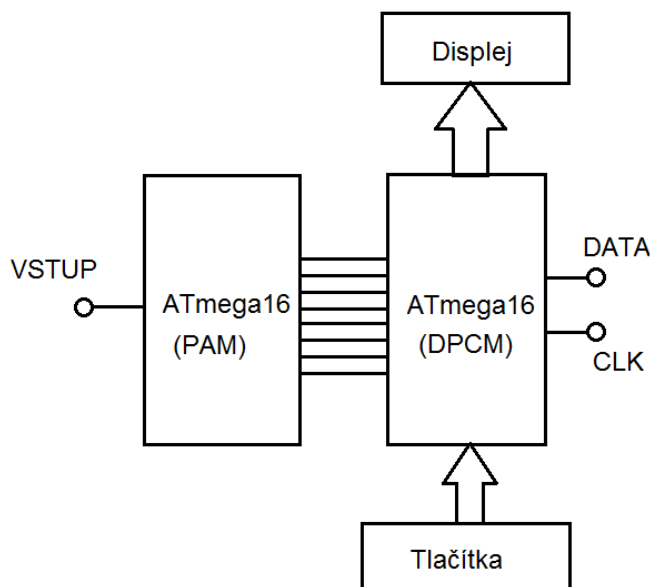


Obrázek č.55: Časové průběhy vstupního signálu modulátoru PCM (oranžový) a výstupního signálu demodulátoru PCM (modrý)

5.2 Programování Diferenční Pulzně Kódové Modulace (DPCM)

Modulátor DPCM

Druhou impulzovou modulací, která byla v rámci této diplomové práce programována je modulace DPCM. Jak již bylo napsáno v kapitole 3.2, jedná se o modulaci vycházející z PCM modulace, která se od ní liší tím, že nepřenáší skutečné hodnoty vzorků, ale pouze rozdíly vzorků po sobě jdoucích. Na obrázku č.56 je naznačeno zjednodušené blokové schéma modulátoru DPCM.



Obrázek č.56: Zjednodušené blokové schéma modulátoru DPCM

Jak je z tohoto obrázku vidět, je blokové schéma modulátoru PCM téměř shodné s blokovým schématem modulátoru DPCM. Základní rozdíl tedy není možné najít zde, ale až v kódu programu realizujícího funkci modulace DPCM. Hlavní funkce tohoto programu je znázorněna níže.

```
int main (void)
{
    InitMasterSPI();           //inicializace funkce SPI
    DDRA = 0x00;              //nastavení portu A na vstupní

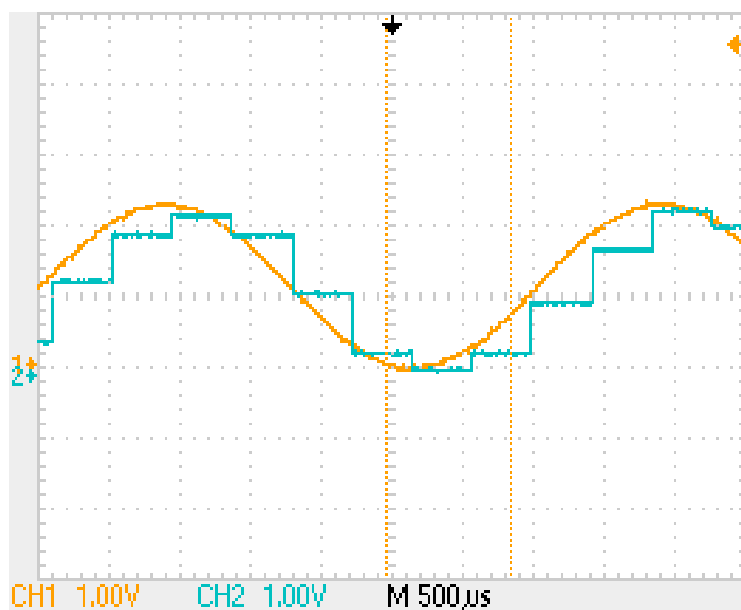
    while(1)                  //nekonečná smyčka programu
    {
        aktualni = PINA;      //načtení hodnoty z A/D převodníku
        ADCx = aktualni - stary; //výpočet rozdílu hodnot
        stary = aktualni;     //aktuální hodnota se nastaví jako stará

        if(ADCx<0)            //pokud nastane situace, že je rozdíl záporný
        {
            ADCx = 128+ ADCx; //zvýší se jeho hodnota o 128
        }
        SPI_MasterTransmit(ADCx); //přenos rozdílu po rozhraní SPI
    }
    return 0;
}
```

Z výše uvedeného kódu aplikace realizující modulaci DPCM je zřejmé, že tato aplikace využívá k přenosu SPI rozhraní. Obdobně jako u modulace PCM je prvním krokem

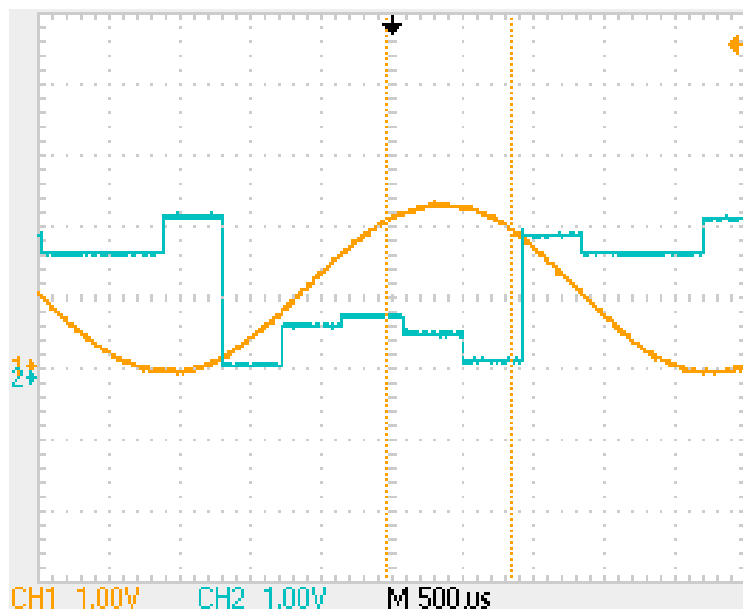
inicializace tohoto rozhraní a nastavení vstupních a výstupních portů mikrokontroléru. Dále se v nekonečné smyčce provádí načtení aktuální hodnoty z A/D převodníku připojeného ke vstupnímu portu A. Z principu modulace DPCM vyplývá, že je uskutečňován přenos rozdílů po sobě následujících hodnot vstupního signálu. Jeho hodnota je v kódu uložena do proměnné $ADCx$. Nedílným krokem je také nastavení aktuální hodnoty do proměnné reprezentující starou hodnotu. Protože při výpočtu rozdílu může nastat situace, kdy jeho hodnota bude záporná, je v kódu umístěna podmínka, která tuto skutečnost kontroluje, a v případě, že tato situace nastane, přičte k rozdílu hodnotu 128. Tato hodnota platí ovšem jen pro 8bitovou DPCM modulaci. Pro případ 4bitové modulace bude tato hodnota rovna 8 a pro případ 2bitové DPCM modulace bude rovna 2. Takto upravená hodnota rozdílu je předána jako vstupní parametr funkce realizující přenos přes SPI rozhraní.

Na následujících obrázcích jsou znázorněny časové průběhy signálů v modulátoru DPCM.



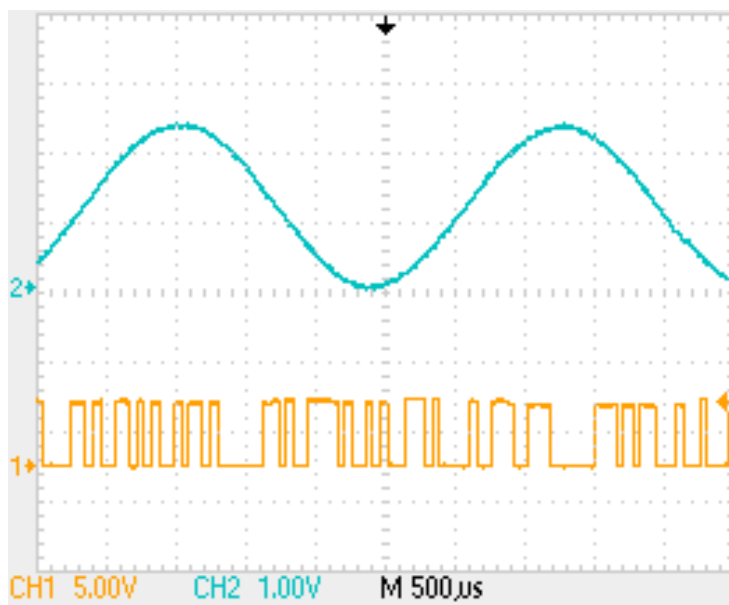
Obrázek č.57: Časové průběhy vstupního signálu a signálu PAM před jeho úpravou v modulátoru DPCM

Na obrázku č.57 jsou zachyceny časové průběhy vstupního signálu (oranžově) a výstupního PAM signálu (modře), který odpovídá aktuálním hodnotám vzorků vstupního signálu, tedy PAM signálu v modulaci PCM. Protože však modulace DPCM nepřenáší tyto hodnoty, ale pouze jejich rozdíly, jsou na následujícím obrázku znázorněny tytéž časové průběhy, avšak s výstupním signálem PAM odpovídající modulaci DPCM.



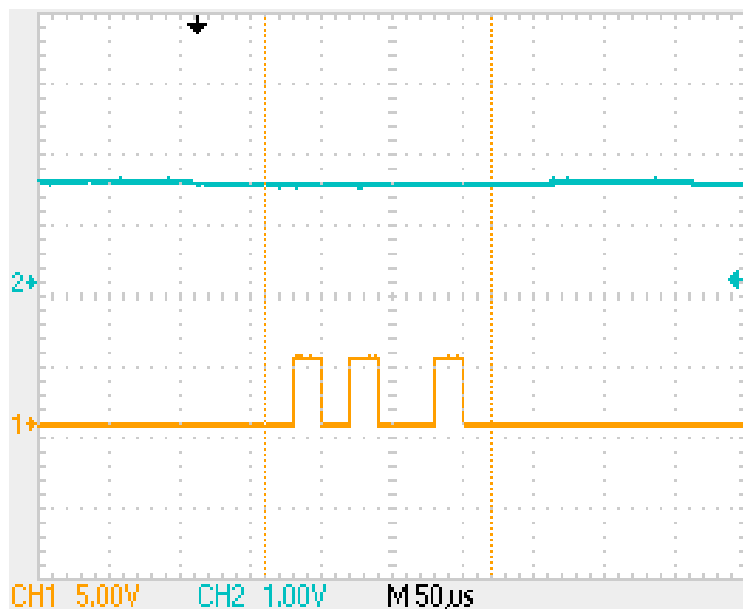
Obrázek č.58: Časové průběhy vstupního signálu a signálu PAM po jeho úpravě v modulátoru DPCM

Na obrázku č.58 jsou zachyceny časové průběhy vstupního signálu (oranžově) a výstupního PAM signálu (modře), který odpovídá rozdílu aktuálních a předchozích vzorků. Jak je zřejmé ze středové části obrázku, jsou tyto rozdíly vždy menší než aktuální hodnota vstupního signálu. Části obrázku, v nichž je zachycen větší skok, odpovídají hodnotám, ke kterým byla přičtena hodnota 128 (pro 8bitovou DPCM). Tato operace byla popsána v jednom z předcházejících odstavců.



Obrázek č.59: Časové průběhy vstupního a výstupního signálu v modulátoru DPCM

Na obrázku č.59 jsou zachyceny časové průběhy vstupního a výstupního signálu v modulátoru DPCM. Pro všechny tři vyobrazené časové průběhy byla vstupní frekvence signálu rovna přibližně 400Hz

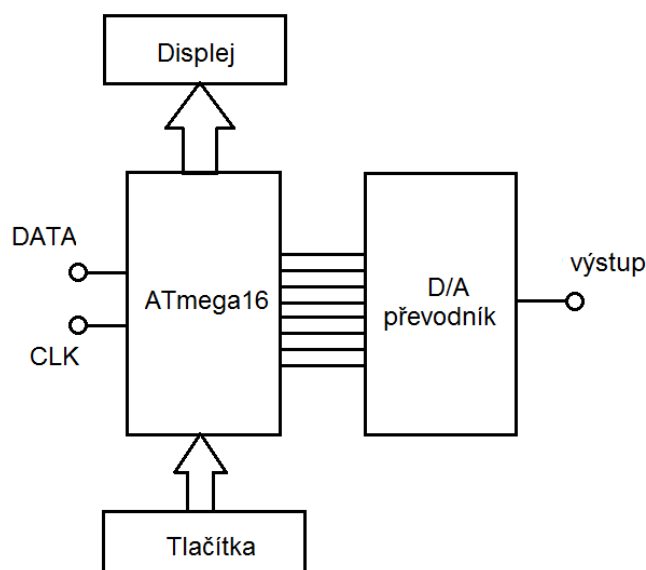


Obrázek č.60: Časové průběhy vstupního konstantního signálu a výstupního signálu demodulátoru

Na obrázku č.60 je zachycen časový průběh výstupního signálu modulátoru DPCM, který je jeho odezvou na konstantní vstupní signál. Jak je z něj patrné, v okamžiku spuštění modulátoru, který je na tomto obrázku označen levým kurzorem, se provede jedna operace, jejímž výstupem je kódová posloupnost. Při dalším cyklu je výstupní kódová posloupnost rovna nule, protože se přenáší hodnota rozdílu aktuálního vzorku vstupního signálu, která je v situaci zachycené na obrázku rovna 1,6V, a minulého vzorku vstupního signálu. Protože se jedná o konstantní signál, je tento rozdíl roven nule.

Demodulátor DPCM

Na obrázku č.61 je naznačeno zjednodušené blokové schéma demodulátoru DPCM. Tohle schéma je téměř totožné s blokovým schématem demodulátoru PCM. Opět se skládá z mikrokontroléru ATmega16, realizující demodulaci DPCM, z převodníku D/A, displeje a tlačítek pro nastavení demodulátoru.



Obrázek č.61: Zjednodušené blokové schéma demodulátoru DPCM

Rozdíl mezi demodulátorem PCM a demodulátorem DPCM není z blokového schématu patrný. Jejich rozdíl je v kódu aplikace realizující funkce obou demodulátorů.

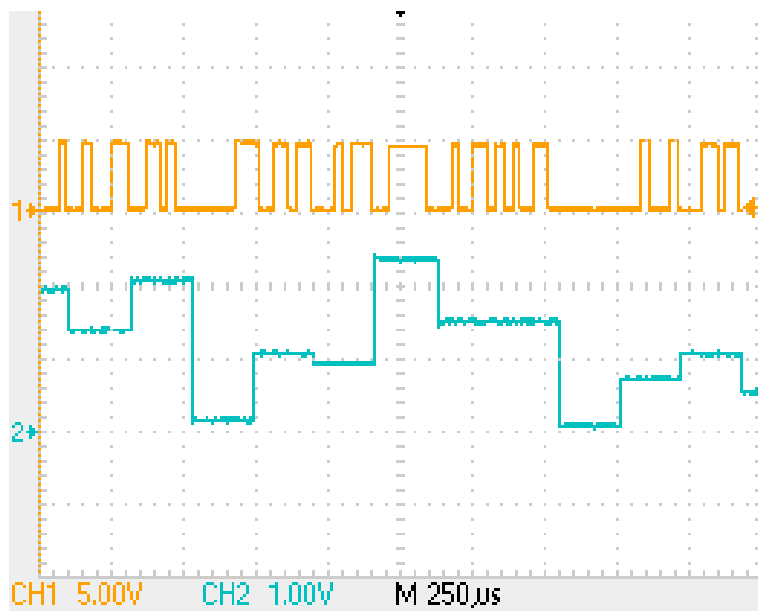
```
int main (void)
{
    InitSlaveSPI();           //inicializace SPI a nastavení režimu SLAVE
    DDRA = 0xff;              //nastavení portu A na výstupní

    while(1)
    {
        if((PINB & 0b10000000)!=0) //čekání na příchod 1.logické 1
        {
            while(1)           //začne se vykonávat nekonečná smyčka
            {
                SPI_SlaveReceive(); //ve které se přijmou data ze sběrnice SPI
                data = SPDR;
                if(data>=128)      //jsou-li tato data větší nebo rovna 128
                {
                    data = data - 128; //musí od nich být odečtena hodnota 128
                }
                PORTA = data;      //následně se pošlou na výstupní port C
            }
        }
    }
    return 0;
}
```

Uvedený kód aplikace, realizující funkci demodulátoru DPCM, je velmi podobný kódu aplikace, realizující funkci demodulátoru u modulace PCM. Jejich zásadním rozdílem je přenášená hodnota. Zatímco, u PCM byla přenášena hodnota aktuálního vzorku, u modulace DPCM je přenášen pouze rozdíl hodnoty aktuálního vzorku a hodnoty minulého vzorku. Po přijetí hodnoty přes rozhraní SPI, dochází u demodulátoru DPCM k ověření, zda tato hodnota

není rovna nebo větší hodnotě 128. V případě, že tomu tak je, musí být tato přijatá hodnota právě o hodnotu 128 snížena. Tato operace je dána inverzní operací prováděnou v modulátoru. Po kontrole a případné úpravě přijaté hodnoty, je tato poslána na výstupní port A, odkud s ní pracuje D/A převodník.

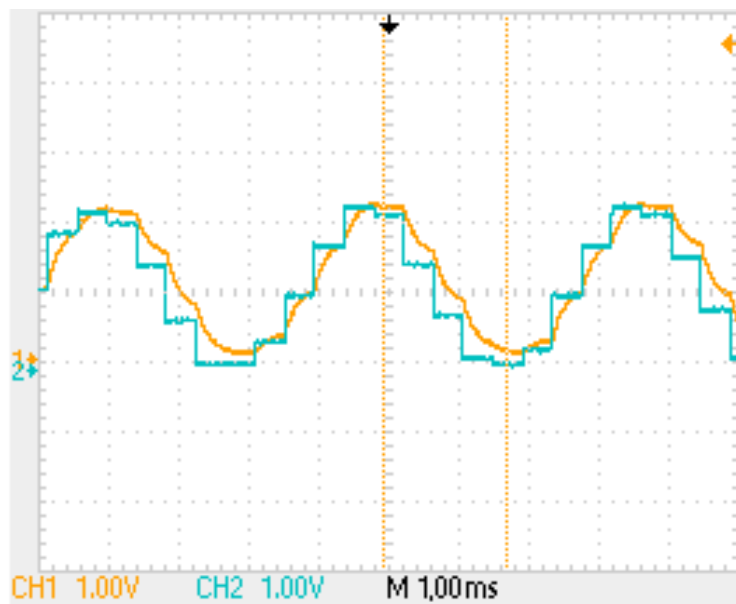
Na následujících obrázcích jsou zachyceny časové průběhy v demodulátoru DPCM.



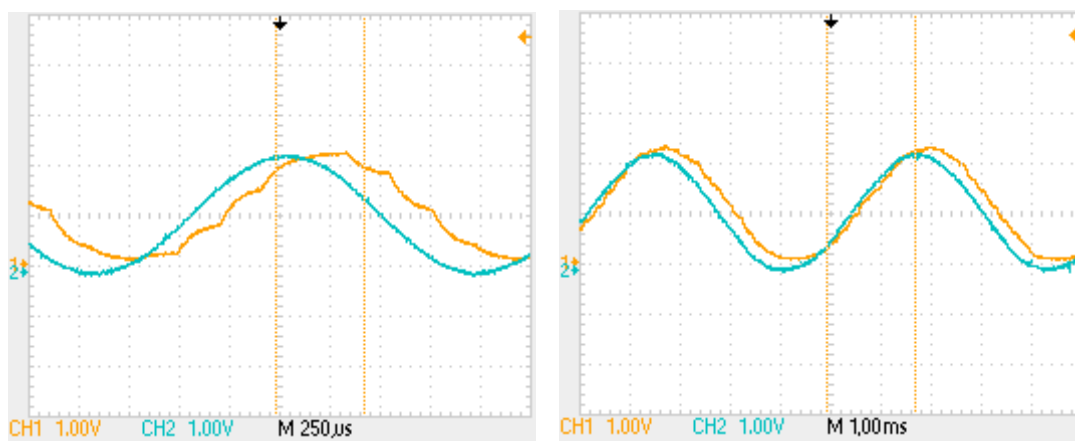
Obrázek č.62: Časové průběhy vstupního signálu a signálu reprezentující rozdíl mezi dvěma po sobě následujícími vzorky v demodulátoru DPCM

Na obrázku č.62 jsou zachyceny časové průběhy vstupního signálu demodulátoru DPCM (oranžový) a signálu reprezentující rozdíl mezi dvěma po sobě jdoucími vzorky v demodulátoru DPCM. Bohužel není kvůli malému zpoždění vznikajícímu v programu demodulátoru možno rozpoznat, kde začíná osmice bitů reprezentující přenášený rozdíl. Kdyby tu tato možnost byla, šlo by jednoznačně ukázat bit reprezentující hodnotu 2^7 , čili 128, která demodulátoru říká, že se aktuálně přijímaná hodnota bude odečítat a nikoli přijímat, jako je tomu v případě, kdy bit reprezentující hodnotu 2^7 je roven nule.

Na dalším obrázku jsou znázorněny časové průběhy PAM signálu, který byl poskládán z přenášených rozdílů a výstupního signálu po filtraci. Jak je možno z tohoto i následujícího obrázku zjistit, při frekvenci přibližně 500Hz není výstupní signál demodulátoru tak hladký jako by byl při frekvenci např. 200Hz.



Obrázek č.63: Časové průběhy PAM signálu po úpravě a PAM signálu po jeho filtraci v demodulátoru DPCM



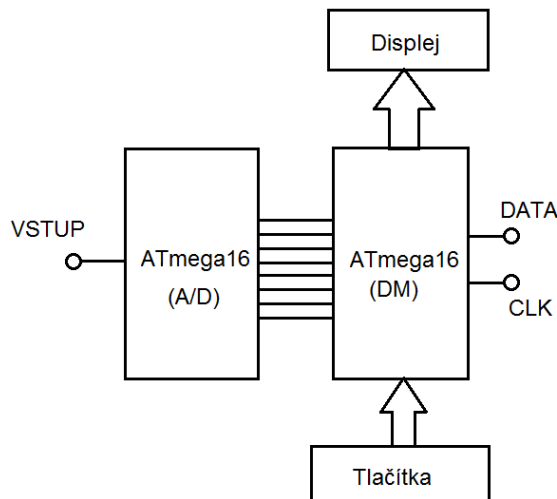
Obrázek č.64: Časové průběhy vstupního signálu modulátoru a výstupního signálu demodulátoru DPCM pro 500Hz a 200Hz

5.3 Programování Delta Modulace (DM)

Třetí impulzovou modulací, která byla v rámci řešení této diplomové práce programována je modulace Delta.

Modulátor delta

Blokové schéma zapojení modulátoru je zobrazeno na obrázku č.65. Obdobně jako v případě modulátorů PCM a DPCM se zapojení skládá z A/D převodníku realizovaného jedním mikrokontrolérem ATmega16 a druhého mikrokontroléru ATmega16, realizujícího delta modulaci. Tohle zapojení je zvoleno kvůli eliminaci zpoždění daného A/D převodem.

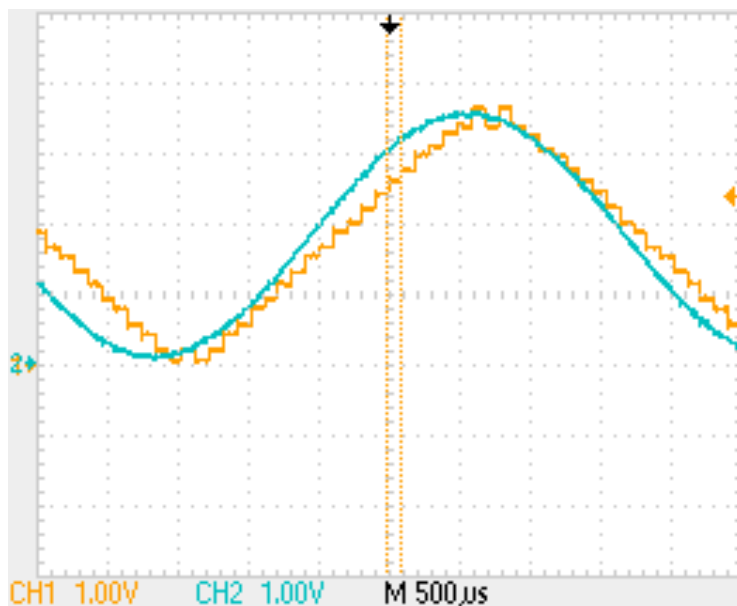


Obrázek č.65: Zjednodušené blokové schéma modulátoru delta modulace

Jak bylo popsáno v kapitole 3.3, je princip této modulace založen na porovnávání aktuální hodnoty vstupního signálu s hodnotou vstupního signálu z předchozího kroku. Část programu realizující funkci modulátoru delta je zobrazena níže.

```
int Delta (void)                                //aplikace realizující DM
{
    DDRB = 0xff;                                //nastavení portu B na výstupní
    DDRD = 0x00;                                //nastavení portu D na vstupní
    while(1)                                    //nekonečná smyčka programu
    {
        PORTB |= (1<<7);                       //nastavení výstupního pinu B7 do log.1
        aktualni = PIND;                        //načtení aktuální hodnoty z AD převodníku
                                                //připojeného ke vstupnímu portu D
        if(aktualni>rozhodovaci)                //je-li aktuální hodnota větší než
        {                                       //rozhodovací
            rozhodovaci += vyska_schodu;        //zvětší se rozhodovací úroveň o
                                                //výšku schodu
            PORTB = 0b00100000;                //na výstupní pin se nastaví logická 1
        }
        else                                    //v opačném případě
        {
            rozhodovaci -= vyska_schodu;        //se sníží rozhodovací úroveň o
                                                //výšku schodu
            PORTB = 0b00000000;                //na výstupní pin se nastaví logická 0
        }
        wait_delta();                           //doby vysílání výstupního bitu
    }
    return 0;
}
```

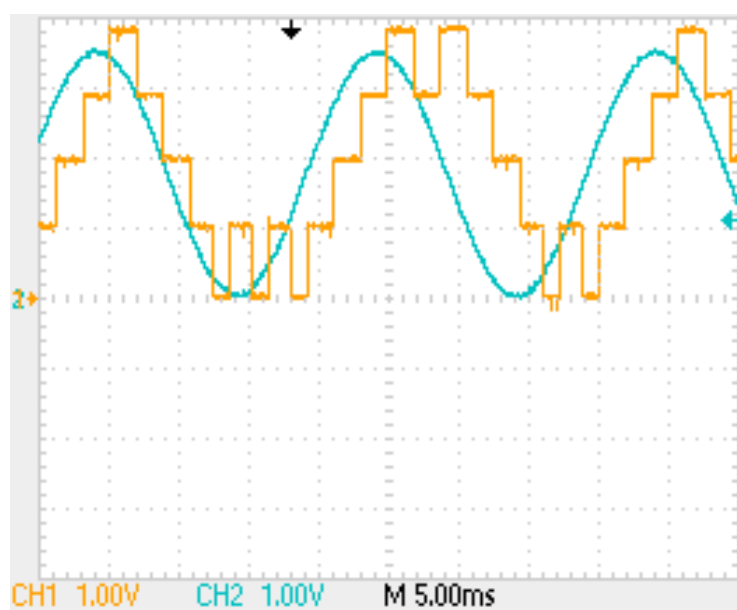
Jelikož je modulátor realizován s externím A/D převodníkem, začíná proces modulace načtením hodnoty z A/D převodníku, který je připojen ke vstupnímu portu D. Zároveň se tato hodnota ukládá do proměnné *aktualni*, dále se tato hodnota porovná s hodnotou uloženou v proměnné *rozhodovaci*, pokud je *aktualni* větší, zvýší se hodnota *proměnné rozhodovaci* o výšku schodu, a výstupní pin B5 je nastaven na hodnotu logické jedničky. V opačném případě, tedy je-li hodnota v *proměnné aktualni* menší nebo rovna hodnotě v *proměnné rozhodovaci*, sníží se hodnota *proměnné rozhodovaci* o výšku schodu, a výstupní pin B5 je nastaven na hodnotu logické nuly. Doba trvání bitu ve výstupním signálu je dána funkcí `wait_delta`.



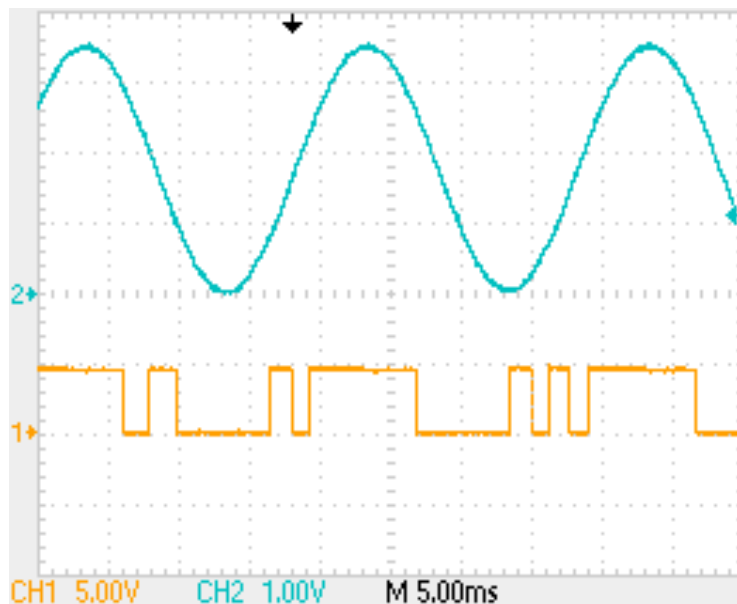
Obrázek č.66: Časové průběhy signálů v modulátoru delta s přetížením kodéru

Na obrázku č.66 je naznačen modrou barvou vstupní signál a oranžovou barvou signál reprezentující úroveň rozhodovacího signálu. V tomto případě byla zvolena malá výška schodu, což má za následek vznik přetížení kodéru.

Na dalším obrázku je naznačen obdobný časový průběh, avšak s větší výškou schodu, která zajistí, že přetížení kodéru vznikat nebude.



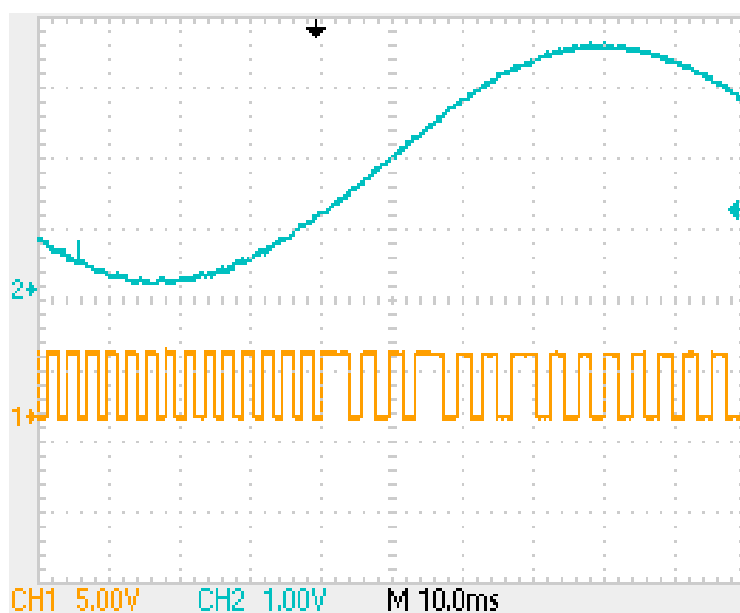
Obrázek č. 67: Časové průběhy signálů v modulátoru delta bez přetížení kodéru



Obrázek č.68: Časové průběhy vstupního a výstupního signálu v modulátoru delta

Na předchozím obrázku jsou naznačeny časové průběhy vstupního (modrý) a výstupního (oranžový) signálu v modulátoru delta. Bohužel z něj není díky skutečnosti, že jako typ výstupního signálu byl oproti simulacím zvolen unipolární signál NRZ.

Na dalším obrázku je znázorněn detail vstupního a výstupního signálu pro nízkou frekvenci.

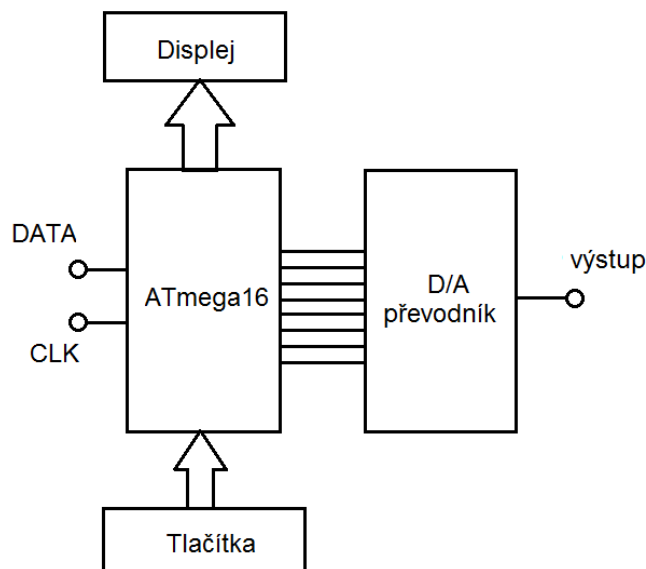


Obrázek č.69: Detail časových průběhů vstupního a výstupního signálu

Z tohoto obrázku je pěkně vidět, že velmi často při nízkých kmitočtech vstupního signálu, dochází k tomu, že výstupní signál pouze mění svou hodnotu z logické jedničky na logickou nulu a obráceně, a pouze zřídka, z širšího pohledu, nastane situace, že po sobě následují dva nebo více pulzů s logickou hodnotou jedna či nula. Tato situace, která však není nikterak na škodu, lze řešit buď snížením vzorkovací frekvence, nebo snížením velikosti výšky schodu.

Demodulátor delta

Demodulátor delta má blokové schéma totožné s demodulátorem PCM i DPCM. Tohle blokové schéma je zobrazeno na následujícím obrázku.



Obrázek č.70: Zjednodušené blokové schéma demodulátoru delta

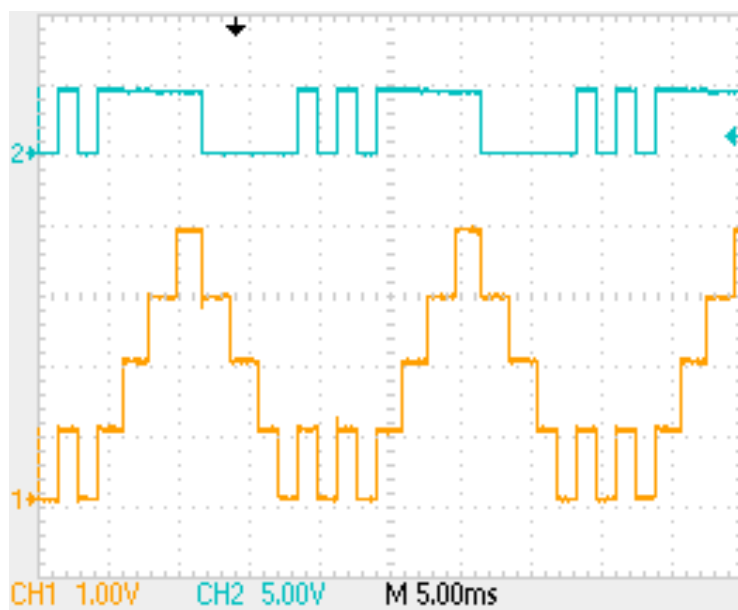
Kód programu realizující demodulaci delta je zobrazen níže.

```
void DELTAdem() //funkce demodulátoru delta
{
    int vystup = 0; //inicializace proměnných
    int data = 0;
    int vyska_schodu = 50;

    while(1)
    {
        if((PINB & 0b10000000)!=0) //čekání na příchod 1.logické 1
        { //potom začíná
            while(1) //nekonečná smyčka programu
            {
                if ((PINB & 0b00100000)== 0b00100000) //zjištění stavu na PINu B5
                { //je-li roven 1
                    vystup +=vyska_schodu; //výstup se zvýší o výšku schodu
                }
                else //pokud není roven 1
                {
                    vystup -= vyska_schodu; //výstup se sníží o výšku schodu
                }
                if(vystup <0) //pokud je výstup menší než 0
                {
                    vystup =0; //nastaví se jeho hodnota na 0
                }
                if(vystup > 255) //pokud je výstup větší než 255
                {
                    vystup = 255; //nastaví se jeho hodnota na 255
                }
                PORTA = vystup; //nastavení výstupu na port A
                wait_delta(); //doba trvání jednoho bitu
            }
        }
    }
}
```

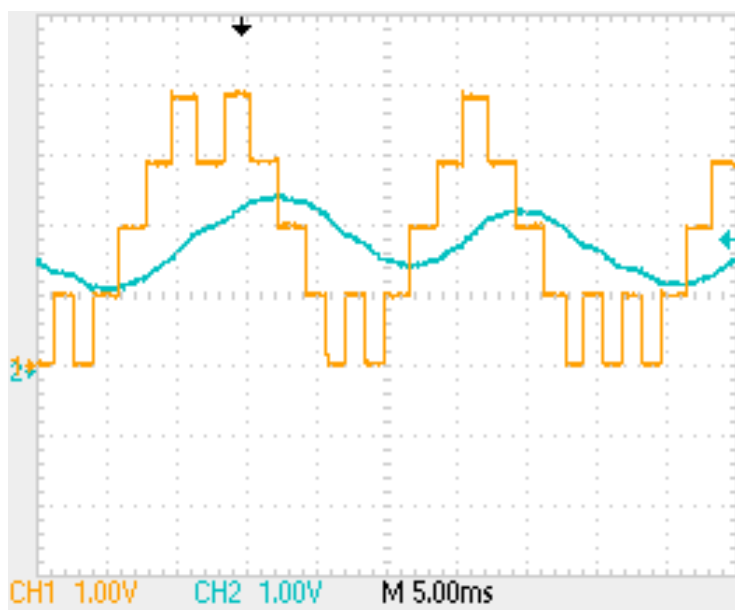
Z uvedeného kódu je patrné, že stejně jako předcházející demodulátory, také demodulátor delta čeká na příjem prvního synchronizačního bitu, po jehož příjmu se dostává do nekonečné smyčky, v níž vykonává několik podmínek. První z nich je podmínka, kterou demodulátor zjišťuje hodnotu na vstupním pinu B5. Je-li v okamžiku vykonávání této podmínky na pinu logická hodnota 1, zvýší se výstupní hodnota o výšku schodu. V opačném případě se výstupní hodnota o výšku chodu sníží. Díky těmto operacím by se mohly nastat stavy, při kterých bude výstupní hodnota vyšší než 255, případně nižší než 0. K zamezení těchto stavů jsou v kódu nastaveny jisté mantinely, které zaručí, že výstupní hodnota nikdy neopustí tento rozsah. Po úplné úpravě výstupní hodnoty ji demodulátor opět umístí na svůj výstupní port A.

Na následujícím obrázku jsou znázorněny časové průběhy signálů v demodulátoru.



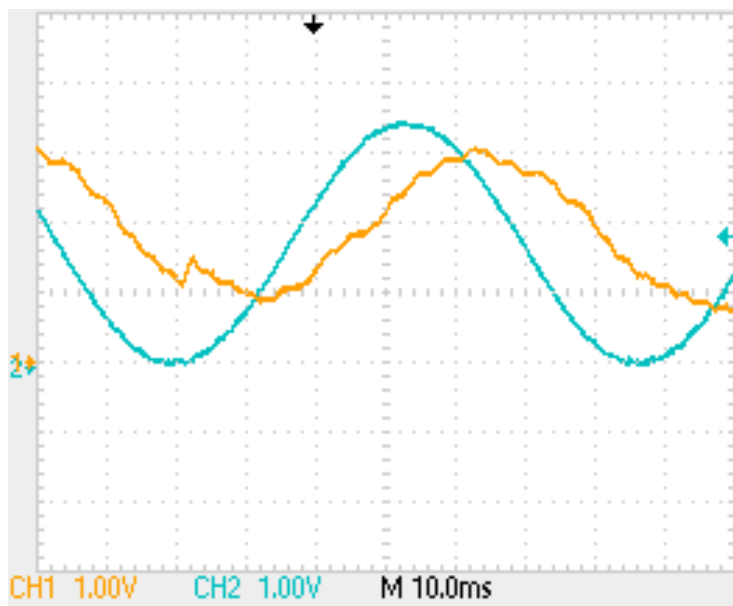
Obrázek č.71: Časové průběhy signálů v demodulátoru delta

Na obrázku č.71 jsou zachyceny dva časové průběhy. První z nich (modrý) reprezentuje vstupní signál demodulátoru, druhý reprezentuje již jakousi formu výstupního signálu. Na následujícím obrázku jsou znázorněny časové průběhy výstupního signálu před filtrací a po filtraci.



Obrázek č.72: časové průběhy výstupního signálu před filtrací a po filtraci

Jak je na tomto obrázku vidět, schodovitý signál již velmi blízce připomíná sinusovku. Na dalším obrázku jsou znázorněny časové průběhy vstupního signálu modulátoru a výstupního signálu demodulátoru.



Obrázek č.73: Časové průběhy vstupního signálu modulátoru a výstupního signálu demodulátoru

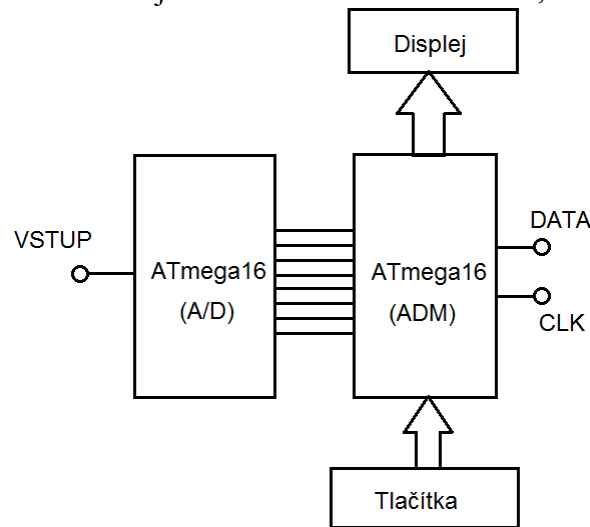
Jak je z obrázku č.73 patrné, je vstupní signál modulátoru (modrý) odlišný od výstupního signálu demodulátoru (oranžový). Tento rozdíl je dán zmiňovaným rekonstrukčním filtrem. Díky tomu, že v zapojení byl použit rekonstrukční filtr druhého řádu, nemohlo nikdy dojít k úplnému vyhlazení výstupního signálu.

5.4 Programování Adaptabilní Delta Modulace (ADM)

Čtvrtou a zároveň poslední impulzovou modulací, která byla v rámci řešení této diplomové práce programována, je adaptabilní delta modulace.

Modulátor ADM

Blokové schéma modulátoru ADM je shodné s modulátorem DM, PCM i DPCM.



Obrázek č.74: Zjednodušené blokové schéma modulátoru ADM

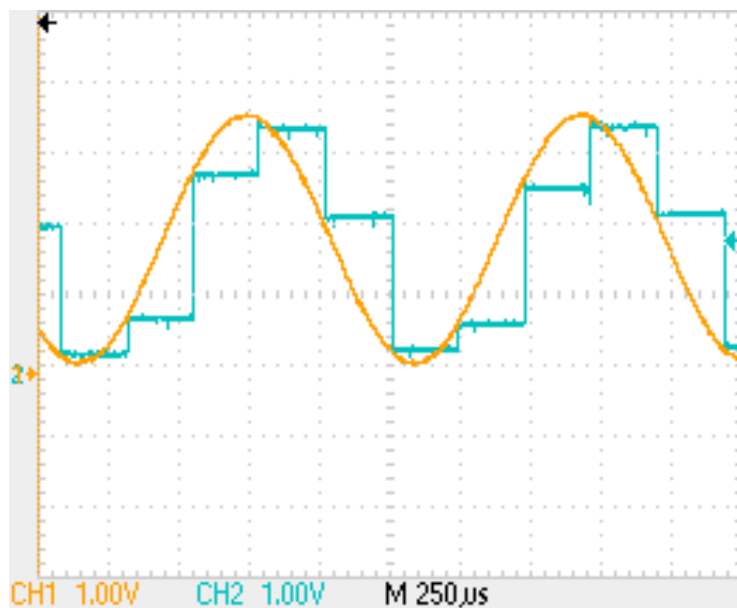
Protože adaptabilní delta modulace principiálně vychází z klasické modulace delta, jsou si kódy programů realizující obě tyto modulace velmi podobné. Začátky obou programů jsou téměř identické, ale u ADM je navíc zavedena tzv. adaptace. Zkrácený kód realizující modulaci ADM je zobrazen níže.

```
void ADM (ADCx)
{
    while(1)
    {
        .           //průběh klasické delta modulace
        .
        .           //začátek funkce adaptace
        if((vystup_stary == vystup) & vyska_schodu<80)
        /*pokud jsou 2 po sobě jdoucí výstupní hodnoty stejné, a zároveň ještě
        nebylo dosaženo maximální hodnoty výšky schodu*/
        {
            vyska_schodu = 2 * vyska_schodu; //zdvojnásobí se výška schodu
        }
        if((vystup_stary != vystup) & vyska_schodu >1)
        /*jsou-li hodnoty dvou po sobě jdoucích výstupních signálů různé, a zároveň
        ještě nebylo dosaženo minimální hodnoty výšky schodu*/
        {
            vyska_schodu = vyska_schodu/2; //zmenší se výška schodu na polovinu
        }
        if(rozhodovaci < 0)           //pokud je rozhodovací úroveň nižší než 0
        {
            rozhodovaci = 0;         //nastaví se na hodnotu 0
        }
        vystup_stary = vystup;        //aktuální výstup se nastaví jako starý
        wait_delta();                 //doba trvání výstupního bitu
    }
}
```

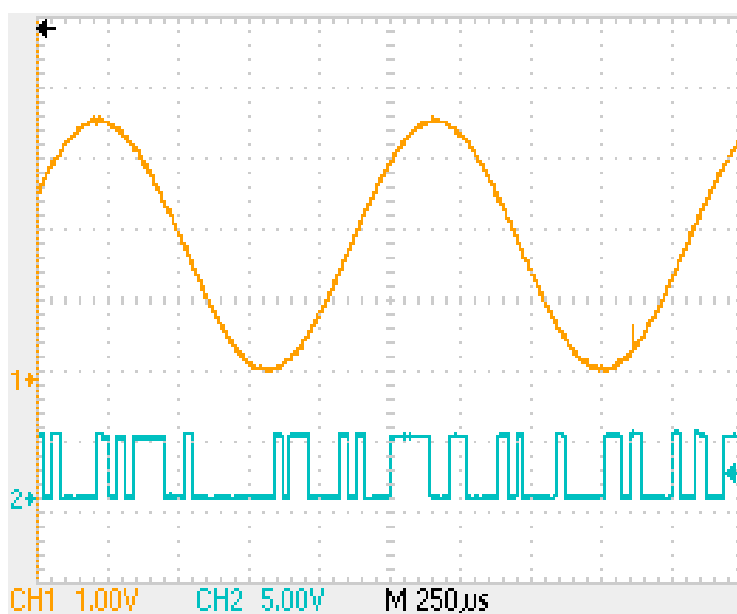
Jelikož byl program realizující modulaci delta popsán v jednom z předcházejících odstavců a začátek programu realizující adaptabilní delta modulaci je stejný jako zmiňovaný kód, bude následovat pouze popis programu zabývající se tzv. adaptací.

Nastane-li případ, že hodnota výstupního signálu je shodná s hodnotou předchozího výstupního signálu, dojde ke zdvojnásobení výšky schodu. V opačném případě, tedy dojde-li k situaci, že hodnota výstupního signálu je opačná vůči hodnotě předchozího výstupního signálu, dojde ke snížení výšky schodu na polovinu. Tento postup vychází z tzv. Winklerova algoritmu. [3]

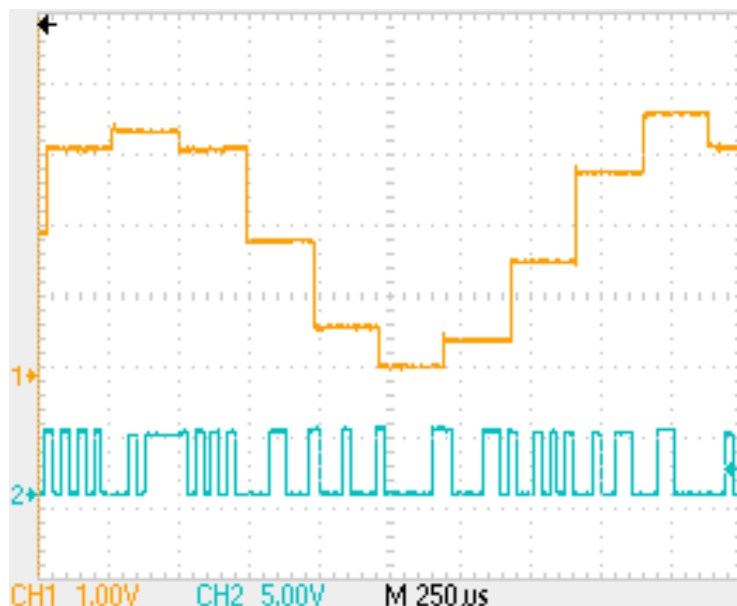
Na následujících obrázcích jsou zobrazeny časové průběhy signálů v modulátoru ADM.



Obrázek č.75: Časové průběhy vstupního signálu a úrovně rozhodovacího signálu v modulátoru ADM



Obrázek č.76: Časové průběhy vstupního a výstupního signálu v modulátoru ADM



Obrázek č.77: Časové průběhy úrovně rozhodovacího signálu a výstupního signálu v modulátoru ADM

Na obrázku č.75 je modrou barvou zobrazen časový průběh vstupního signálu a oranžovou barvou časový průběh signálu reprezentující úroveň rozhodovacího signálu. Jak je z něj patrné mění se výška schodu v každém kroku.

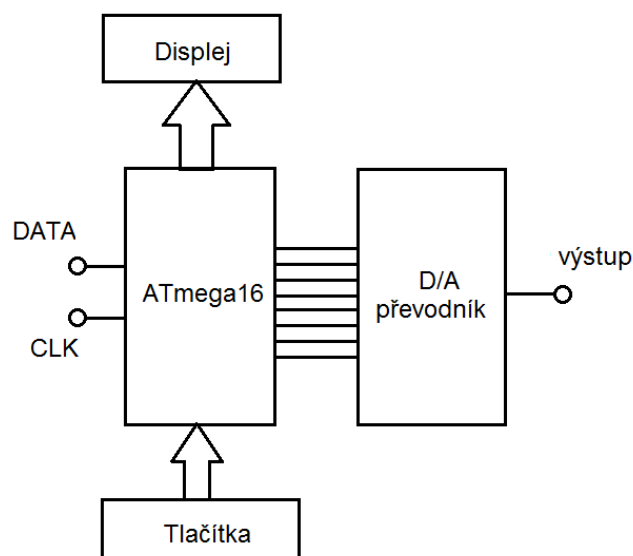
Na obrázku č.76 jsou zachyceny časové průběhy vstupního signálu (oranžový) a výstupního signálu (modrý) modulátoru ADM.

Na obrázku č.77 jsou zachyceny časové průběhy signálu reprezentujícího úroveň rozhodovacího signálu (oranžový) a výstupního signálu modulátoru ADM.

Všechny tyto obrázky odpovídají vstupnímu signálu o frekvenci přibližně 850Hz.

Demodulátor ADM

Zjednodušené blokové schéma demodulátoru adaptabilní delta modulace odpovídá blokovému schématu demodulátoru delta modulace a je zobrazeno na obrázku č.78.



Obrázek č.78: Zjednodušené blokové schéma demodulátoru adaptabilní delta

Při návrhu demodulátoru ADM dochází k menší komplikaci, kterou je proměnlivá výška schodu. V rámci simulací prováděných v prostředí Matlab-Simulink, byla výška schodu přenášena přímo od modulátoru k demodulátoru. V případě programování aplikací, bylo vzhledem k požadavkům na laboratorní přípravky nutné, zajistit změnu výšky schodu u demodulátoru jinou cestou. Došlo tedy k rozšíření programu o tzv. zpětnou adaptaci [2]. Tato zpětná adaptace je principiálně shodná s adaptací použitou v modulátoru. Kód programu realizující demodulátor adaptabilní delta modulace je zobrazen níže.

```
void ADMdem() //funkce demodulátoru delta
{
    vystup = 0; //inicializace proměnných
    vstup_stary = 0;
    vstup = 0;
    int vyska_schodu = 50;

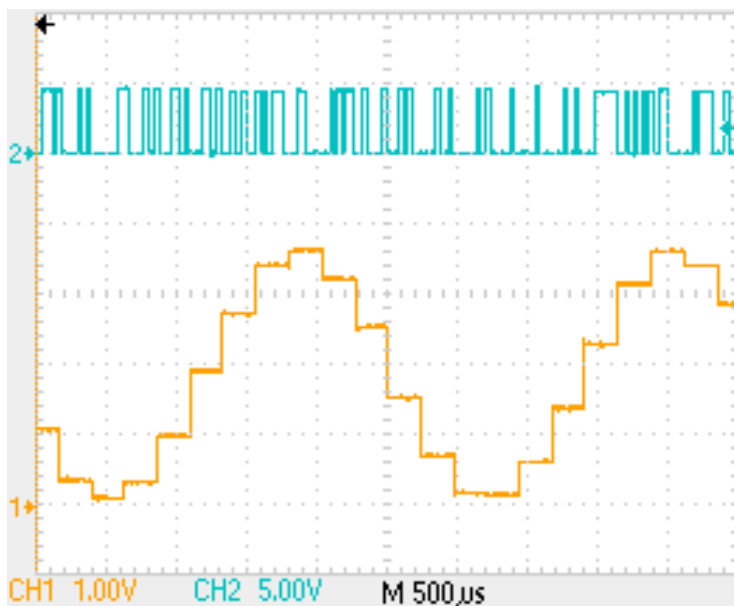
    while(1)
    {
        if((PINB & 0b10000000)!=0) //čekání na příchod 1.logické 1
        { //potom začíná
            while(1) //nekonečná smyčka programu
            {
                if ((PINB & 0b00100000)== 0b00100000) //zjištění stavu na PINu B5
                { //je-li roven 1
                    vstup = 1; //nastaví se proměnná vstup na hodnotu 1
                    vystup +=vyska_schodu; //výstup se zvýší o výšku schodu
                }
                else //pokud není roven 1
                {
                    vstup = 0; //nastaví se proměnná vstup na hodnotu 0
                    vystup -= vyska_schodu; //výstup se sníží o výšku schodu
                }
                if(vystup < 0) //pokud je výstup menší než 0
                {
                    vystup = 0; //nastaví se jeho hodnota na 0
                }
                if(vystup > 255) //pokud je výstup větší než 255
                {
                    vystup = 255; //nastaví se jeho hodnota na 255
                }
                PORTA = vystup; //nastavení výstupu na port A
                if((vstup == vstup_stary)&vyska_schodu<80)
                {
                    vyska_schodu = 2 * vyska_schodu;
                }
                if((vstup != vstup_stary)&vyska_schodu>1)
                {
                    vyska_schodu = vyska_schodu;
                }

                wait_delta(); //doba trvání jednoho bitu
            }
        }
    }
}
```

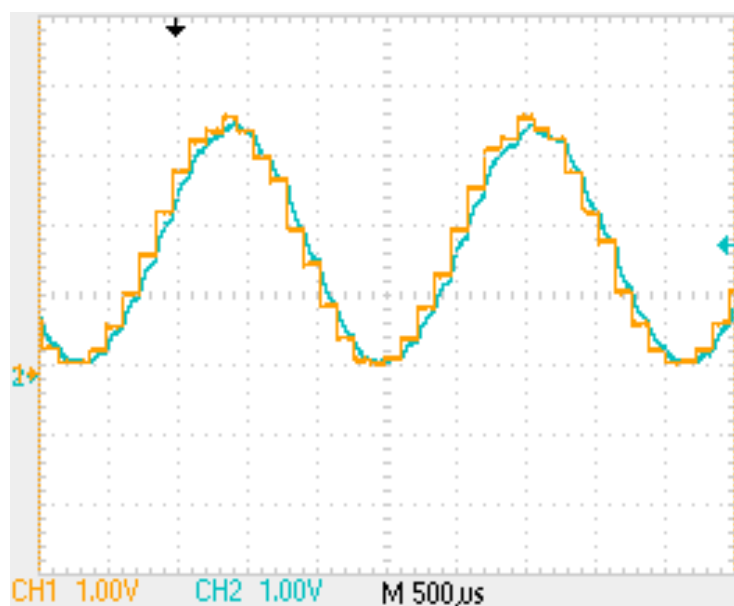
Stejně jako tomu bylo doposud u demodulátorů všech programovaných impulzových modulací, začíná i v tomto případě kód programu synchronizací, spočívající v čekání na první příchodí hodinový impulz. Po jeho přijetí, se program dostává do nekonečné smyčky, v níž provádí demodulaci delta, stejným způsobem, jaký byl popsán v kapitole týkající se právě tohoto demodulátoru. Na konci každého cyklu dochází ke zpětné adaptaci, která podle

aktuální přijaté hodnoty vstupního signálu a hodnoty přijatého vstupního signálu z minulého cyklu, upraví výšku schodu. Na úplném konci každého cyklu je program pozdržen po dobu odpovídající příjmu jednoho bitu.

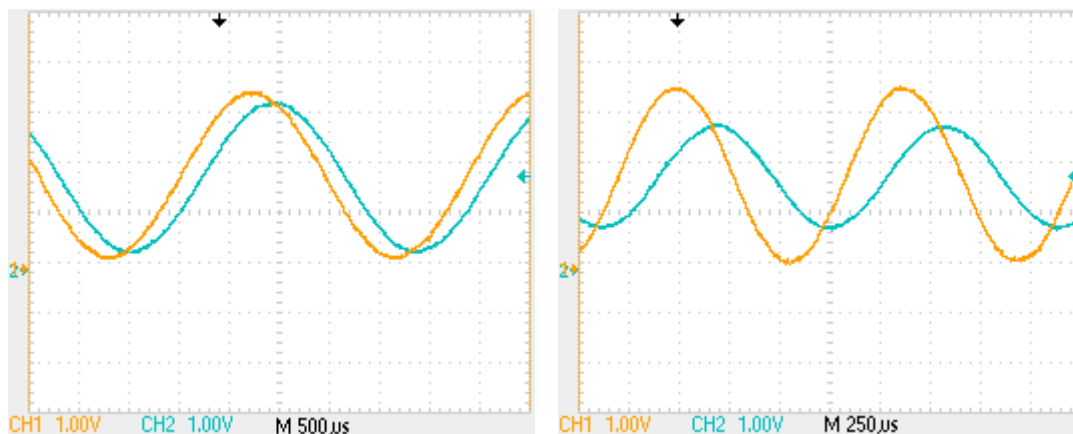
Na následujících obrázcích jsou znázorněny časové průběhy signálů v demodulátoru.



Obrázek č.79: Časové průběhy vstupního a výstupního signálu před filtrací, v demodulátoru ADM



Obrázek č.80: Časové průběhy výstupního signálu demodulátoru před filtrací a po filtraci

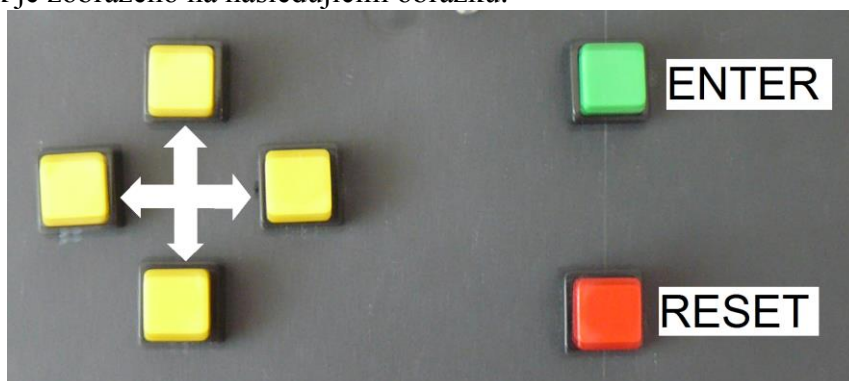


Obrázek č.81: Časové průběhy vstupního signálu modulátoru a výstupního signálu demodulátoru ADM

5.5 Programování obslužné aplikace pro jednotlivé přípravky

Protože cílem práce nebylo pouze naprogramovat jednotlivé typy modulací, ale také zhotovit laboratorní přípravky, realizující tyto modulační procesy, bylo nutné vytvořit jistou obslužnou aplikaci, díky níž, bude uživatel moci obsluhovat tyto přípravky. Jak je patrné z koncepce popsané v kapitole 7, každý přípravek je složen z procesoru, vykonávajícího samotnou modulaci/demodulaci, z bloků vstupů a výstupů, z bloků předzpracování signálu a dodatečného zpracování signálu a v neposlední řadě také z bloku řízení a bloku zobrazení. Právě tyto dva posledně zmíněné bloky, a zejména jejich programování, budou v následujících odstavcích popsány.

Blok řízení umožňuje uživateli pomocí tlačítek provádět výběr modulační metody a nastavení jejích parametrů. K tomu bylo potřeba naprogramovat funkci jednotlivých tlačítek, kterých je na přípravku celkem šest. Čtyři z nich jsou směrová tlačítka (žlutá barva), jedno plní funkci tlačítka ENTER (zelená barva) a poslední funguje jako RESET (červená barva). Rozložení těchto tlačítek je zobrazeno na následujícím obrázku.



Obrázek č.82: Rozložení tlačítek na zařízení

Tlačítka nahoru/dolů mění v obslužné aplikaci vlastnost vybrané modulační metody, tlačítka vlevo/vpravo, mění hodnotu zvoleného parametru. Tlačítko ENTER provádí odsouhlasení aktuálního parametru a tlačítko RESET, ruší vše, aplikace tedy začíná od začátku.

Části kódu programu pro obsluhu jsou naznačeny níže. Jako první je naznačena funkce obsluhující jednotlivá tlačítka.

```
void tlacitko_vlevo ()           //funkce obsluhující tlačítka vlevo
{
    if ((PINC &0b00000001)==0)  //ověření stisknutí daného tlačítka
    {                             //je-li stisknuto
        poloha++;                //zvýší se hodnota proměnné o 1
        _delay_ms(100);          //ochrana proti vícenásobnému stisknutí
        if(poloha == max_poloha+1) //po překročení maximální hodnoty
        {
            poloha = 1;          //proměnná poloha, je nastavena na 1
        }
        enter = false;          //proměnná enter je nastavena na false
    }
}
```

Každé z tlačítek nahoru, dolů, vlevo, vpravo a ENTER jsou připojena k jednotlivým pinům vstupního portu C mikrokontroléru. Po stisku daného tlačítka se provedou operace pro něj definované.

Pro tlačítka *vlevo* se inkrementuje hodnota uložená v proměnné *poloha*, která reprezentuje hodnotu vybraného parametru. Tatáž operace se provede i po stisku tlačítka *vpravo*, jen s tou změnou, že se hodnota uložená v proměnné *poloha* dekrementuje. Při stisku tlačítka *nahoru/dolů* se změní nastavovaný parametr.

Pro upřesnění, parametrem se rozumí např. vzorkovací frekvence a hodnotou tohoto parametru může být 8kHz, 4kHz atd.

Tlačítka RESET je připojeno k pinu RESET na mikrokontroléru.

Po stisknutí tlačítka ENTER je uložena aktuální hodnota proměnné *poloha* a parametr uloží do řídicí struktury, která je páteří celého procesu nastavení modulace.

```
typedef struct rizeni {int modulace,... //definice řídicí struktury
... bit_vzorek, fvz, vyska_schodu;}TRizeni; //s parametry

TRizeni Ridici = {0, 0, 0, 0};          //inicializace proměnné Ridici
```

Kód definice této struktury je zobrazen výše. Parametry definované v této řídicí struktuře, jejich význam a počet možných hodnot jednotlivých parametrů jsou uvedeny v tabulce 7.2

Tabulka č.5.1: Parametry řídicí struktury

Označení parametru	Význam parametru	Počet hodnot parametru
modulace	Druh zvolené modulace	4
bit_vzorek	Počet bitů/vzorek pro modulace PCM a DPCM	3
fvz	Vzorkovací frekvence	6
vyska_schodu	Výška schodu, pro DM a ADM	3

Příklad kódu, realizující uložení hodnot do proměnné `Ridici` po stisknutí tlačítka ENTER je uveden níže. Tento příklad zobrazuje situaci, kdy má uživatel vybránu modulaci PCM, vzorkovací frekvenci, a nyní vybírá počet bitů na vzorek.

```
if (parametr ==3)           //výběr na kolik bitů bude provedena kvantizace
{
    max_poloha = 3;         //jsou 3 možnosti
    switch(poloha)
    {
        case 1:             //1. možností je 8bitů/vzorek
            line_2();        //zobrazení aktuální volby na 2.řádku
            lcd_text("< 8bit/vzorek >"); //displeje
            if(enter ==true) //pokud bude při tomto výběru stisknuto
            {               //tlačítko ENTER,
                Ridici.bit_vzorek = 8; //do proměnné Ridici se na pozici bit/vz
            }               //umístí hodnota 8
            break;
            .
            .
            .
            .
    }
    enter = false;
}
```

Jak je naznačeno v předcházejícím kódu, volba parametrů je realizována pomocí funkce `switch`. Pro lepší přehlednost je zde uvedena pouze jedna možnost.

V následující části kapitoly je popsán výběr modulace a nastavení jejích parametrů, doplněný o snímky LCD displeje.

Po zapnutí zařízení, případně po jeho restartování vypadá řídicí struktura následovně.

parametr	typ modulace	počet bitů/vzorek	vzorkovací frekvence	výška schodu
hodnota	0	0	0	0

Hodnoty všech parametrů jsou nastaveny na nulu. Aby mohlo zařízení správně vykonávat svoji činnost, musí být hodnoty všech parametrů nastaveny na korektní hodnoty. Prvním krokem nastavení je výběr modulace.

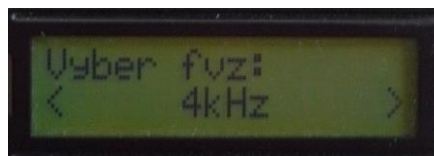


Obrázek č.83: Výběr modulace

Na obrázku č.83, je zobrazen displej zařízení, při kroku, kdy uživatel provádí výběr modulace, kterou má dané zařízení (modulátor/demulátor) provádět. Tlačítka vlevo a vpravo si vybírá druh modulace. Po nalezení požadované ji vybere stisknutím tlačítka ENTER, což zařídí, že se typ dané modulace zapíše na příslušnou pozici do řídicí struktury. Po této volbě tedy řídicí struktura vypadá následovně.

parametr	typ modulace	počet bitů/vzorek	vzorkovací frekvence	výška schodu
hodnota	1	0	0	0

Pro případ výběru modulace DPCM do políčka `modulace` ve struktuře nastavena hodnota jedna, ale hodnoty ostatních parametrů zůstávají nulové. V dalším kroku uživatel provádí výběr vzorkovací frekvence.

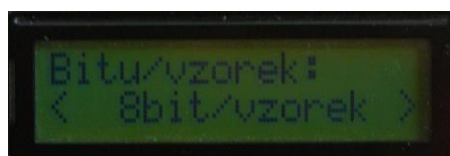


Obrázek č.84: Výběr vzorkovací frekvence

Na obrázku č.84 je zobrazen displej zařízení, při výběru vzorkovací frekvence. Obdobně jako při výběru typu modulace, také v tomto případě je možno tlačítka vlevo a vpravo vybrat jednu ze tří nabízených vzorkovacích frekvencí. Konečný výběr je proveden stisknutím tlačítka ENTER. Po této volbě vypadá řídicí struktura následovně.

parametr	typ modulace	počet bitů/vzorek	vzorkovací frekvence	výška schodu
hodnota	1	0	4000	0

Pro zobrazený případ, v němž je vybrána vzorkovací frekvence 4kHz, se do příslušného políčka nastaví hodnota 4000. Hodnoty ostatních parametrů zůstávají nezměněny. Ve třetím kroku uživatel provádí výběr počtu bitů na vzorek.

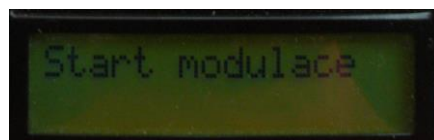


Obrázek č.85: Výběr počtu bitů na vzorek

Na obrázku č.85 je zobrazena situace, kdy je uživatelem vybírán počet bitů na vzorek. Opět je možné vybírat ze tří možností. Stejně jako v přechodících dvou případech, je konečný výběr proveden stisknutím tlačítka ENTER. Po této poslední volbě parametrů modulace vypadá řídicí struktura následovně

parametr	typ modulace	počet bitů/vzorek	vzorkovací frekvence	výška schodu
hodnota	1	8	4000	0

Volbou dalšího parametru, tedy stiskem tlačítka dolů, se dostává uživatel ke startu modulace. Na následujícím obrázku je ukázán stav displeje, odpovídající zmíněné situaci.



Obrázek č.86: Spuštění modulace

Až do této chvíle bylo možné se vrátit k jakémukoliv parametru a provést jeho přenastavení. Po stisku tlačítka ENTER je ukončeno nastavování parametrů a začíná samotný proces modulace a jakékoliv přenastavení je možné pouze po restartu zařízení.

Blok zobrazení je tvořen LCD displejem, který odráží pokyny uživatele zadané tlačítky. Programy a funkce, díky nimž je možno obsluhovat LCD displej byly použity ze zdroje [5].

6. Porovnání simulací impulzových modulací a měření jejich naprogramovaných verzí

V této kapitole se práce věnuje především diskuzi výsledků získaných měřeními naprogramovaných modulací.

Porovnání modulace PCM

Při simulaci modulace PCM bylo dosaženo výsledků, které byly popsány v kapitole 4. Protože při programování modulace PCM byly použity dva návrhové vzory, které se od sebe mírně odlišovaly, budou v následujících odstavcích nejprve porovnány tyto dva mezi sebou.

Modulace PCM využívající pro přenos SPI rozhraní.

Jako velká výhoda použití rozhraní SPI, při programování PCM, se jevila jednoduchost synchronizace modulátoru a demodulátoru.

Drobnou nevýhodou použití tohoto rozhraní, mohla být nemožnost nastavení přesného časového intervalu, během kterého byl výstupní signál v logické hodnotě H nebo L. Tento interval byl nastavován pomocí předděličky frekvence oscilátoru, jejíž hodnotu bylo možno nastavit pouze na omezené množství hodnot. Protože nejvyšší vzorkovací frekvence při modulaci PCM měla být 8kHz, bylo této děličky s výhodou použito.

Pro zmiňovaný příklad platilo, že doba trvání výstupního pulzu pro 8bitovou modulaci PCM měla být 15,625μs.

Při frekvenci krystalu 8MHz a s použitím předděličky /128 byla výsledná hodnota frekvence výstupních pulzů 62,5kHz. Při této frekvenci byla doba trvání jednoho výstupního pulzu 16μs. Tím tedy vznikla nepřesnost, která se nejvíce projevila ve snížení vzorkovací frekvence z 8kHz na 7,8125kHz. Tím se také snížila mezní frekvence ze 4kHz na 3,9kHz.

Další komplikací, při použití rozhraní SPI, byl jiný počet kvantovacích bitů nežli zmiňovaných 8. Tento problém vyvstal z principu rozhraní SPI, které vždy přenáší 8bitové slovo. Řešení tohoto problému bylo následovné. Při nastavení s požadavkem na 4 kvantovací bity, bylo nutno každý výstupní bit poslat dvakrát po sobě.

Příklad 8bitového slova obsahující hodnotu 170 pro

a) 8bitový přenos

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	0	1	0

b) 4bitový přenos

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	0	0	1	1	0	0

c) 2bitový přenos

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	1	0	0	0	0

V případě, že by nebyla tato operace uskutečněna, docházelo by ke zkreslení. Protože by přijatá slova vypadala následovně

a) 8bitový přenos

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	0	1	0

b) 4bitový přenos

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	1	0	1	0

c) 2bitový přenos

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	0	1	0

Přenesené hodnoty by tedy vůbec neodpovídaly hodnotám získaným z A/D převodníku a celá modulace by byla znehodnocena.

Modulace PCM nevyužívající pro přenos SPI rozhraní.

Všechny popsané nevýhody v modulaci PCM, která pro přenos využívala SPI rozhraní byly velmi jednoduše vyřešeny v modulaci PCM, která rozhraní SPI pro přenos nepoužívala. Její největší výhodou bylo, že každý bit výstupního signálu posílala ihned na výstupní pin, tudíž jí bylo úplně jedno, zda se jednalo o 8bitovou, nebo 2bitovou PCM modulaci.

Nevýhodou tohoto řešení však mohla být synchronizace modulátoru s demodulátorem. Každý z nich totiž generoval časový interval reprezentující dobu trvání výstupního/vstupního impulsu nezávisle na sobě. To by v případě i malého rozdílu ve frekvenci krystalů obou zařízení mohlo mít vliv na synchronizaci.

Porovnání simulací a měření naprogramovaných modulací

Největším rozdílem mezi simulacemi a měřením impulzových modulací na naprogramovaném zařízení byl výstupní signál demodulátoru. Zatímco při měření impulzových modulací se projevoval rekonstrukční filtr druhého řádu jistým zkreslením výstupního signálu demodulátoru, u simulací bylo možné dosáhnout lepších výsledků zejména díky možnosti použití filtru vyššího řádu.

Dalším rozdílem mezi simulacemi a měřením byla mezní frekvence. Opět u simulací bylo možno dosáhnout teoretické mezní frekvence, tedy poloviny frekvence vzorkovací, při měření byla mezní frekvence vždy o něco nižší než její teoretická hodnota. Při použití modulátoru pracující s rozhraním SPI, byla mezní frekvence téměř totožná s teoretickou, ale při použití modulátoru nevyužívající rozhraní SPI se reálná mezní frekvence od teoretické již více lišila.

Tabulka č.6.1: Dosažené mezní frekvence pro modulaci PCM

Teoretická	4kHz	2kHz	1kHz
Simulace	4kHz	2kHz	1kHz
Měření s SPI	3,86kHz	1,92kHz	993Hz
Měření bez SPI	3,34kHz	1,6kHz	930Hz

Porovnání modulace DPCM

Při simulaci modulace DPCM bylo obdobně jako u simulace PCM možno dosáhnout lepších výsledků nežli při měření naprogramované modulace. Programování modulace DPCM bylo na rozdíl od programování modulace PCM realizováno pouze s využitím rozhraní SPI. Díky této skutečnosti bylo možno dosáhnout lepších mezní frekvencí, které byly srovnatelné s výsledky měření pro modulaci PCM využívající rozhraní SPI.

Tabulka č.6.2: Dosažené mezní frekvence pro modulaci DPCM

Teoretická	640Hz	320Hz	160Hz
Simulace	640Hz	320Hz	160Hz
Měření s SPI	638Hz	315Hz	153Hz

Porovnání modulace DM

Při programování modulace delta nebylo na rozdíl od přechozích případů využíváno rozhraní SPI. Byl používán stejný mechanismus, jako při modulaci PCM nevyužívající rozhraní SPI. Co se týče mezních frekvencí, ty byly při simulacích i měření téměř totožné. Malá odchylka, která u nich vznikla, byla dána zpoždovací smyčkou programu, kde nebylo možné nastavit úplně přesný časový interval. I přes tuto malou odchylku byly výsledné mezní frekvence na velmi dobré úrovni.

Tabulka č.6.3: Dosažené vybrané mezní frekvence pro modulaci DPCM

Teoretická	2kHz	4kHz
Simulace	2kHz	4kHz
Měření	1989Hz	3,992kHz

Porovnání modulace ADM

Stejně jako při programování DM, také k programování ADM byl použit vzor bez využití rozhraní SPI. Obdobně jako v předchozím případě, také u měření modulace ADM bylo dosaženo výsledků srovnatelných se simulacemi.

Jak již bylo uvedeno u porovnání modulace PCM, největším rozdílem bylo možno sledovat na výstupním signálu demodulátoru. Opět díky nízkému řádu filtru byl výstupní signál mírně zkreslen oproti vstupnímu.

Při nastavení minimální hranice výšky schodu na hodnotu, odpovídající přibližně dvacetině maximální hodnoty vstupního signálu a maximální hranice výšky schodu na hodnotu, odpovídající přibližně jedné třetině maximální hodnoty vstupního signálu, bylo dosaženo nejlepších výsledků jak z pohledu mezní frekvence, tak z pohledu odchylky výstupního signálu demodulátoru od vstupního signálu modulátoru.

Co se týče mezních frekvencí, ty byly podobně jako u modulace delta na velmi dobré úrovni v porovnání se simulacemi. V případě použití zmíněného nastavení hranic výšky schodu, a použití vzorkovací frekvence 128kHz, byla pro signál s amplitudou 2,5V, podle vztahu (8), mezní frekvence rovna 6,5kHz. Při měření této modulace, bylo dosaženo mezní frekvence 6487Hz.

7. Návrh koncepce pro výrobu demonstračních přípravků

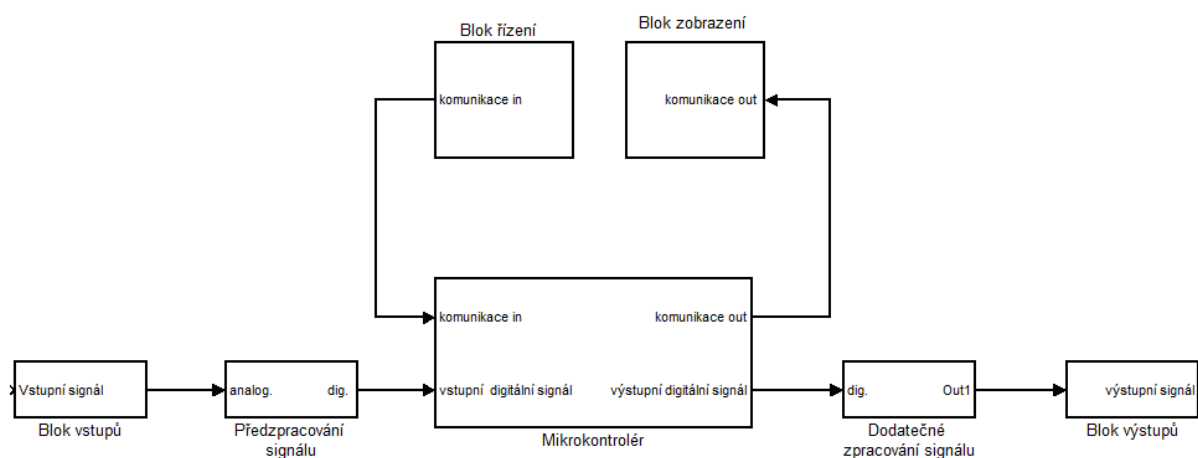
Cílem tohoto návrhu bylo vytvoření jisté koncepce pro výrobu sady laboratorních přípravků modulátor – demodulátor. U modulátoru byla požadavkem možnost volby typu použité modulace a jejích parametrů. Požadavek na demodulátor byl stejný. Po správném nastavení těchto parametrů by měl modulátor a demodulátor provádět správně zvolenou modulaci.

Jelikož bylo cílem práce porovnat mezi sebou jednotlivé druhy modulací a jejich parametry, bylo nutné, aby modulátor uměl alespoň dvě modulace, které jsou v principu úplně odlišné, a alespoň dvě modulace, které jsou si v principu podobné, a jejich výsledky závisí na parametrech modulátoru. Tohoto cíle bylo dosaženo modulacemi PCM, DPCM, DM a ADM.

Jádrem obou přípravků byl mikrokontrolér ATmega16, který měl v zapojení realizovat samotnou modulaci. Jelikož tento mikrokontrolér obsahuje A/D převodník, nebylo třeba do zapojení umisťovat jiný obvod provádějící A/D převod. Po prvních měřeních naprogramovaných modulací však bylo zjištěno, že samotný A/D převod realizovaný přímo v mikrokontroléru provádějším zvolenou modulaci, má příliš velké zpoždění. Z tohoto důvodu bylo rozhodnuto o doplnění zapojení o druhý mikrokontrolér provádějící pouze A/D převod.

Z důvodu absence D/A převodníku v mikrokontroléru, bylo nutné zapojení realizující demodulátor o něj doplnit. Jako vhodný obvod byl zvolen DAC0832, provádějící 8bitový převod D/A.

Na obrázku č.41 je zobrazena koncepce jednoho modulátoru/demodulátoru.



Obrázek č.87: Koncepce modulátor/demodulátor

8. Návrh zapojení a výroba laboratorních přípravků

V této kapitole se práce věnuje popisu návrhu a výroby laboratorních přípravků. Při tomto návrhu vychází z koncepce uvedené v předcházející kapitole.

Dalším z cílů práce byl návrh schémat a výroba laboratorních přípravků realizující popsané modulace. Návrh schémat modulátor i demodulátoru vycházel z koncepce přípravků uvedené v kapitole 7. Každý z těchto přípravků se tedy skládal z mikrokontroléru realizující zvolenou modulaci, blok řízení, blok zobrazení, blok předzpracování a blok dodatečného zpracování signálu.

Návrh zapojení přípravku modulátoru

Zapojení modulátoru bylo s ohledem na problémy se zpožděním daným dobou trvání A/D převodu, popsané v kapitole 5.1, realizováno jako modulátor s externím A/D převodníkem. To znamenalo, že v zapojení figurovaly dva mikrokontroléry ATmega16. Jeden plnil pouze funkci A/D převodníku a druhý realizující zvolenou modulaci.

V případě modulací PCM a DPCM byl jedním z požadavků výstupní signál doplněný o výstupní signál modulace PAM. Díky tomuto požadavku bylo nutné také blok modulátoru doplnit o D/A převodník realizovaný integrovaným obvodem DAC0832. Zapojení tohoto integrovaného obvodu je podle [10] realizováno spojením s OZ. Tento OZ vyžaduje symetrické napájení $\pm 12V$. Aby nebylo nutné připojovat k přípravkům velké množství zdrojů, je nutné nějakým způsobem ze zmiňovaného napájení $\pm 12V$ získat napětí o hodnotě 5V. Tato malá komplikace byla vyřešena integrovaným obvodem 7805-STM. Tento lineární regulátor napětí dokáže ze vstupního napětí až 15V udělat výstupní napětí o velikosti 5V.[11]

Návrh zapojení přípravku demodulátoru

Zapojení demodulátoru bylo navrženo velmi obdobně jako zapojení modulátoru. Zásadním rozdílem těchto dvou zapojení byl počet mikrokontrolérů v jednotlivých zapojeních. Jak bylo napsáno v předchozím odstavci, zapojení modulátoru obsahovalo dva mikrokontroléry. Na rozdíl od něj, obsahovalo zapojení demodulátoru mikrokontrolér pouze jeden. Stejně jako tomu bylo u modulátoru, také demodulátor musel být doplněn o integrovaný obvod DAC0832 realizující D/A převodník. Díky tomu, musely být do zapojení přidány operační zesilovače vyžadující symetrické napájení. Naprosto stejným způsobem byl tedy do zapojení přidán regulátor napětí 7805-STM.

Schémat zapojení obou přípravků, návrhy desek plošných spojů, osazovací výkresy a seznam součástek byly přidány do příloh na konec práce.

Dalším krokem byla výroba krabic přípravků. K tomuto účelu velmi dobře posloužily krabice typu U-KP12. Na tyto krabice byly přiděleny zdířky připojení napájení a pro vstupní a výstupní signály. Dále byl přidělen na každou z krabic displej a tlačítka pro komunikaci uživatele se zařízením.

Na následujících obrázcích jsou vyobrazeny finální výrobky.



Obrázek č.88: Finální výrobky

Na následujícím obrázku je vyobrazena zadní strana obsahující zdířky napájení. Jak bylo zmíněno v předcházejícím textu, napájení je symetrické $\pm 12V$.



Obrázek č.89: Pohled na zadní stranu přípravku demodulátoru

9. Vzor výpočtu a tabulky vypočítaných hodnot

V první části této kapitoly se práce zaměřuje na vzory výpočtů jednotlivých parametrů, ve druhé části jsou obsaženy tabulky s vypočítanými hodnotami

1. Vzor výpočtu vzorkovací frekvence pro přenos hlasového signálu pomocí PCM podle (2)

Hlasový signál má rozsah kmitočtů 300Hz až 3400Hz. [2]

$$f_{VZ} \geq 2 \cdot f_{\max} \geq 2 \cdot 3400 \geq \underline{\underline{6800\text{Hz}}}$$

2. Vzor výpočtu počtu kvantizačních hladin pro 8bitovou PCM

$$N_{KV} = 2^{nb} = 2^8 = \underline{\underline{256}}$$

3. Vzor výpočtu kvantizačního kroku pro 8bitovou PCM se vstupním sinusovým signálem s amplitudou 2,5V podle vztahu (4)

$$q = \frac{FS}{N_{KV} - 1} = \frac{5}{256 - 1} = \frac{5}{255} = \underline{\underline{19,6\text{mV}}}$$

4. Vzor výpočtu doby trvání synchronizačního pulzu pro 8bitovou PCM se vzorkovací frekvencí 8kHz podle vztahu (1) a (5)

$$t_b = \frac{T_V}{2 \cdot nb} = \frac{\frac{1}{f_{VZ}}}{2 \cdot nb} = \frac{\frac{1}{8000}}{2 \cdot 8} = \frac{1}{128000} = \underline{\underline{7,8125\mu\text{s}}}$$

5. Vzor výpočtu maximální difference signálu pro sinusový signál s amplitudou 2,5V, maximální frekvencí 500Hz, za předpokladu vzorkovacího signálu 8kHz podle vztahu (6)

$$\Delta_{\max} = 2 \cdot A \cdot \sin\left(2 \cdot \pi \cdot f_{\max} \cdot \frac{1}{2 \cdot f_{VZ}}\right) = 2 \cdot 2,5 \cdot \sin\left(2 \cdot \pi \cdot 500 \cdot \frac{1}{2 \cdot 8000}\right) = \underline{\underline{0,975\text{V}}}$$

6. Vzor výpočtu minimálního počtu bitů DPCM, při uvažování kvantizačního kroku 19,6mV a maximální difference vstupního signálu 1,25V. (7)

$$N = \log_2\left(\frac{2 \cdot \Delta_{\max}}{q}\right) = \log_2\left(\frac{2 \cdot 1,25}{0,0196}\right) = 6,99 \Rightarrow \underline{\underline{7\text{bitů}}}$$

7. Vzor výpočtu mezní frekvence, pro možnost použití 7bitové DPCM namísto 8bitové PCM. Ze vztahu (7) lze určit maximální diferenci rozdílového signálu

$$N = \log_2 \left(\frac{2 \cdot \Delta_{\max}}{q} \right) \Rightarrow \Delta_{\max} = \frac{2^N \cdot q}{2} = \frac{2^7 \cdot 0,0196}{2} = \underline{\underline{1,25V}}$$

Ze vztahu (6) lze vypočítat maximální možný kmitočet pro tuto diferenci

$$\Delta_{\max} = 2 \cdot A \cdot \sin \left(2 \cdot \pi \cdot f_{\max} \cdot \frac{1}{2 \cdot f_{VZ}} \right) \Rightarrow$$

$$f_{\max} = \frac{2 \cdot f_{VZ}}{2 \cdot \pi} \cdot \arcsin \left(\frac{\Delta_{\max}}{2 \cdot A} \right) = \frac{16000}{2 \cdot \pi} \cdot \arcsin \left(\frac{1,25}{2 \cdot 2,5} \right) \approx \underline{\underline{643Hz}}$$

8. Vzor výpočtu mezní frekvence, pro nepřetížení DM kodéru. Vzorkovací signál je uvažován 128kHz, amplituda vstupního signálu je 5V, výška schodu je 0,5V. Ze vztahu (8) lze odvodit vztah pro výpočet mezní frekvence.

$$A \cdot \Omega_{\max} \leq \frac{\Delta s}{T_V} \Rightarrow \Omega_{\max} = \frac{\Delta s}{A \cdot T_V} = \frac{0,5}{5 \cdot \frac{1}{128000}} = 12800 \text{ rad}^{-1}$$

$$\Omega_{\max} = 2 \cdot \pi \cdot f_{\max} \Rightarrow f_{\max} = \frac{\Omega_{\max}}{2 \cdot \pi} = \frac{12800}{2 \cdot \pi} = \underline{\underline{2037Hz}}$$

Tabulky vypočítaných hodnot:

Tabulka č.9.1: Maximální hodnoty kmitočtu vstupního signálu při daném vzorkovací kmitočtu u PCM

Vzorkovací kmitočet [Hz]	Maximální kmitočet vstupního signálu [Hz]
8	4
4	2
2	1

Tabulka č.9.2: Počet kvantizačních úrovní pro daný počet bitů u PCM

Počet bitů PCM	Počet kvantizačních úrovní
8	256
4	16
2	4

Tabulka č.9.3: Velikost kvantizačního kroku pro daný počet bitů u PCM a vstupní signál s rozsahem 5V

Počet bitů PCM	Velikost kvantizačního kroku [V]
8	0,0196
4	0,3333
2	1,6667

Tabulka č.9.4: Doba trvání synchronizačního pulzu pro nbitovou PCM a danou vzorkovací frekvenci

Počet bitů PCM	Vzorkovací frekvence [Hz]	Doba trvání synchronizačního pulzu [μs]
8	8000	7,8125
4	8000	15,6250
2	8000	31,2500
8	4000	15,6250
4	4000	31,2500
2	4000	62,5000
8	2000	31,2500
4	2000	62,5000
2	2000	125,0000

Tabulka č.9.5: Mezní frekvence vstupního signálu pro danou vzorkovací frekvenci, v případě snížení počtu bitů DPCM o 1 bit vůči počtu bitů PCM

Počet bitů PCM	Počet bitů DPCM	Maximální difference [V]	Vzorkovací frekvence [Hz]	Mezní frekvence [Hz]
8	7	1,255	8000	643,05
4	3	1,333	8000	687,73
2	1	1,667	8000	865,83
8	7	1,255	4000	323,18
4	3	1,333	4000	343,86
2	1	1,667	4000	432,91
8	7	1,255	2000	161,59
4	3	1,333	2000	171,93
2	1	1,667	2000	216,46

Tabulka č.9.6: Mezní frekvence vstupního signálu pro danou vzorkovací frekvenci, danou výšku schodu

Výška schodu [V]	Vzorkovací frekvence [kHz]	Mezní frekvence [Hz]
0,1	128	407,64
0,5	128	2037,22
1	128	4076,43
0,1	64	203,82
0,5	64	1019,11
1	64	2038,22
0,1	32	101,91
0,5	23	509,55
1	32	1019,11

Závěr

Impulzové modulace

V první kapitole nazvané Impulzové modulace byl vysvětlen pojem modulace. Dále v ní byl uveden rozdíl mezi analogovou a impulzovou modulací. Nejdůležitější částí této kapitoly však bylo základní rozdělení impulzových modulací na nekvantované a kvantované.

Nekvantované impulzové modulace

V této kapitole byly popsány základní principy nekvantovaných impulzových modulací. U každé z těchto modulací byl slovně popsán princip modulace, který byl doplněn obrázkem ilustrující celou situaci.

Studiem nekvantovaných impulzových modulací, bylo dospěno k závěru, že tyto modulace nejsou příliš vhodné k přenosu dat. Tato skutečnost je dána zejména jejich náchylností k rušení.

Uplatnění nekvantovaných impulzových modulací, zejména pulzně šířkové modulace PWM je v různých oblastech vyjma přenosu dat. Nejčastější uplatnění této modulace je ve výkonové elektrotechnice, především k řízení chodu elektromotorů.

Kvantované impulzové modulace

Této kapitole byl věnován v práci větší prostor. Na začátku kapitoly byly popsány základní principy používané u kvantovaných impulzových modulací. Každá z těchto modulací byla, podobně jako tomu bylo v předchozí kapitole, slovně popsána a tento popis byl také doplněn obrázkem pro lepší pochopení slovního popisu.

Dále byl pro každou z uvedených kvantovaných modulací zhotoven demonstrační model v prostředí Matlab-Simulink a pomocí těchto modelů, byly provedeny simulace jednotlivých modulací s jednotlivými parametry.

Pulzně kódová modulace

Pro tuto modulaci byly vytvořeny celkem tři demonstrační modely. Každý z nich byl vytvořen pro reprezentaci jedné ze tří možností počtu bitů na vzorek. Na výběr tedy byly modely reprezentující 8bitovou, 4bitovou a 2bitovou PCM. U každého z těchto modelů bylo možné nastavit vzorkovací frekvenci na jeden ze tří rozsahů. Těmi byly 8kHz, 4kHz a 2kHz. Přestože se jednalo o nejjednodušší kvantovanou impulzovou modulaci, dosahuje tato modulace velmi dobrých výsledků, jak je patrné z obrázků č.36 a č.37. Kvalita modulace a demodulace byla v podstatě závislá pouze na počtu bitů reprezentující jeden vzorek. Vzorkovací frekvence omezovala tuto modulaci pouze v kmitočtovém rozsahu vstupního signálu.

Oblast použití této impulzové modulace je velmi široká. Její nejčastější použití je v převodnících A/D, kde tvoří základ celého převodu analogového signálu na digitální signál. V hudebním průmyslu je PCM využívána u audio CD, kde provádí převod A/D s vzorkovací frekvencí 44,1kHz a pro reprezentaci jednoho vzorku používá 16 bitů. V telekomunikacích je podobně jako v hudebním průmyslu PCM využívána pro převod A/D, v tomto případě se však používá vzorkovací frekvence 8kHz a pro reprezentaci jednoho vzorku je využito 8 bitů.

Jak již bylo řečeno, PCM je sice nejjednodušší kvantovaná impulzová modulace, ale díky svým vlastnostem je velmi kvalitní.

Diferenční pulzně kódová modulace

Podobně jako pro případ PCM, byly také pro modulaci DPCM vytvořeny 3 modely, které měly stejné parametry, jako modely v případě PCM. Pro přechod mezi DPCM a PCM bylo v práci využito principu, který využíval snížení dynamiky vstupního signálu díky provedení rozdílu mezi po sobě jdoucími prvky. Díky tomuto snížení dynamiky, bylo možné kvantovat na nižší počet bitů než by tomu bylo u PCM, aniž by došlo ke ztrátě informace nebo k velkému zkreslení.

Jak je patrné z obrázků č.38 a č.39 přenosové vlastnosti systému DPCM byly opravdu shodné s přenosovými vlastnostmi PCM, avšak pouze v případě použití 8bitové a 4bitové DPCM. V případě použití 2bitové DPCM, přestával popsáný princip fungovat a docházelo ke zkreslení signálu a tedy i ke ztrátě informace.

I přes tento handicap, je DPCM stejně jako PCM velmi kvalitní. Má však jednu velmi nepříjemnou vlastnost, která nebyla zatím v práci popsána. Jedná se o tzv. kumulaci chyb. Tato největší nevýhoda DPCM vzniká z principu této modulace. Jelikož se při ní kódují pouze rozdíly vzorků, nikoliv jejich celé hodnoty, může se stát, že při přenosu se změní zakódovaná hodnota, a demodulátor nepozná, že nastala chyba. V případě, že by chyba nastala u PCM, projeví se pouze v bezprostředním okolí daného vzorku.

Výhodou oproti PCM je však ušetření přenosové kapacity, kterého je dosaženo kvantováním nižších hodnot. Aby bylo tohle snížení nároku na šířku pásma ještě větší, musely by být vzorky odebírány s vyšší frekvencí, aby se ještě více projevila jejich vzájemná korelace.

DPCM má téměř stejné uplatnění jako PCM. Zmiňované snížení nároků na šířku pásma se projevuje hlavně v oblasti telekomunikací. Dalším odvětvím, kde se využívá vlastností DPCM je zpracování obrazu. V případě velkých ploch se stejnou nebo podobnou barvou a jasem se využívá toho, že rozdíl mezi jednotlivými pixely je velmi nízký, nemusí se tedy kvantovat do tolika úrovní.

Delta modulace

Stejně jako v předchozích dvou případech, také pro modulaci delta byl vytvořen v prostředí Matlab-Simulink demonstrační model, u něhož je možné zvolit si ze tří vzorkovacích frekvencí a ze tří možností výšky schodu.

Na rozdíl od PCM a DPCM, potřebovala DM pro svou činnost mnohem vyšší vzorkovací frekvence. To bylo dáno principem převodu, kdy se kódoval pouze jeden bit na jeden vzorek vstupního signálu. Tento jediný bit určuje, zda vstupní signál roste, nebo klesá.

Z obrázku č.23 je patrné, že výstupní signál nesledoval vstupní signál tak hladce, jako tomu bylo například u 8bitové PCM. Tohle malé zkreslení bylo dáno principem modulace.

Větším problémem, který může u DM vznikat, bylo tzv. přetížení kodéru, tedy situace, kdy aproximační signál nestíhá sledovat rychlé změny vstupního signálu. Tato situace byla zobrazena na obrázku č.24. Tomuto problému lze zabránit například omezením maximální frekvence vstupního signálu.

Nejčastějším využitím DM je podobně jako u všech kvantovaných impulzových modulací, převod A/D a D/A.

Sigma delta modulace

Tak jako v případě DM, byl i pro sigma delta modulaci vytvořen demonstrační model v prostředí Matlab-Simulink. Stejně jako u DM bylo možno si v tomto modelu vybrat jednu ze tří vzorkovacích frekvencí a jednu ze tří možných výšek schodu.

Díky zapojení integrátoru na vstup modulace delta, byla snížena dynamika vstupního signálu. Tímto se velmi snížila pravděpodobnost přetížení kodéru. Jak je zobrazeno na obrázku č.26, snižovala sice SDM dynamiku vstupního signálu, avšak integrátor zanášel do průběhu zpoždění. Díky němu nebyla SMD shledána vhodnou pro práci v systémech s reálným časem.

Podobně jako DM, se také SDM nejčastěji využívá v převodnících A/D a D/A. Tyto převodníky sice nejsou tak rychlé jako v případě převodníků využívající DM, avšak díky tomu, že může být použita nižší vzorkovací frekvence, použít i pomalejší procesor realizující převod.

Adaptabilní delta modulace

Pro ADM byly vytvořeny dva demonstrační modely, pro něž platí to samé jako pro modely DM a SDM. Jeden z nich prováděl ADM bez omezení výšky schodu, druhý s omezením výšky schodu.

Zásadním důvodem k zavedení ADM, je předcházení přetížení kodéru. To se díky oběma těmto modelům podařilo, avšak v případě použití ADM bez omezení výšky schodu, docházelo k situaci, kdy po rychlé změně vstupního signálu docházelo k prudkému zvýšení výšky schodu. Jak bylo naznačeno na obrázku č.40, způsobila tato velká výška schodu na výstupním signálu záškuby, namísto toho, aby výstupní signál sledoval signál vstupní. Tím tedy mohou vznikat velké rozdíly mezi vstupním a výstupním signálem a může docházet ke ztrátě informace.

Tato situace je vyřešena ADM s omezenou výškou schodu. Jak je patrné z obrázku č.40, právě tato modulace dosahovala při simulacích nejlepších výsledků.

Při návrhu demonstračních modelů ADM byl zvolen pro adaptaci Winklerův algoritmus.

V porovnání DM, SDM a ADM dosahuje nejlepších výsledků, co se týče předcházení přetížení kodéru, právě adaptabilní modulace s omezením výšky schodu.

Přes toto prvenství, však není ani tato modulace dokonalá. Vzniká zde totiž problém v podobě přenosu výšky schodu. U klasické DM, je výška schodu stále stejná, proto nemá demodulátor nejmenší problém demodulovat svůj vstupní signál. V případě ADM je však výška schodu v každé periodě jiná a demodulátoru musí být tato hodnota nějakým způsobem sdělována.

V případě simulačních modelů nebylo problémem předávat tuto hodnotu přímo od modulátoru k demodulátoru.

Programování impulzových modulací

Programování impulzových modulací do zařízení s mikrokontrolérem ATmega16 probíhalo podobně jako vytváření demonstračních modelů. Celé programování probíhalo ve vývojovém prostředí AVR studio 4.

Pulzně kódová modulace

Pro pulzně kódovou modulaci bylo vytvořeno hned několik programů, z nichž každý pracoval na trochu jiném principu. Prvním program pracoval na principu, kdy jediný mikrokontrolér realizoval jak A/D převod tak samotnou modulaci. Jak bylo při měření tohoto modulátoru zjištěno, docházelo u něj vlivem zpoždění, které bylo dané dobou trvání A/D převodu, k nežádoucím zpožděním ve výstupním signálu. Tato situace byla zobrazena na obrázku č.42. Proto byl tento princip zavrhnut.

Jeho náhradou byl princip, který využíval zapojení dvou mikrokontrolérů, z nichž jeden realizoval pouze A/D převod a druhý realizoval samotnou modulaci. Z časových průběhů zobrazených na obrázku č.44 je patrné, že se touto cestou podařilo odstranit ono nechťené zpoždění.

Pro tento princip vzorkování byly navrženy dva programy, z nichž každý realizoval přenos od modulátoru k demodulátoru jiným způsobem.

První z těchto programů využíval, pro přenos sériového rozhraní mikrokontroléru SPI. Druhý z těchto programů využíval pro přenos pouhé nastavování výstupního pinu do logických hodnot L nebo H.

Výhody a nevýhody obou těchto principů byly diskutovány v kapitole 6. Z této diskuse vyplývá, že vhodnějším principem bude ten, využívající pro přenos rozhraní SPI, zejména pro jednodušší synchronizaci modulátoru a demodulátoru.

Diferenční pulzně kódová modulace

Po programování modulace PCM bylo na řadě programování modulace DPCM. Z poznatků získaných z programování předchozí modulace, byla tato programována s externím A/D převodníkem. Co se týče principu přenosu signálu od modulátoru k demodulátoru, bylo zde využito SPI rozhraní. Stejně jako tomu bylo u simulací DPCM, také při programování a následném měření bylo zjištěno, že modulace funguje pouze jako 8bitová a 4bitová. V případě 2bitové modulace, již docházelo k obrovskému zkreslení výstupního signálu vůči signálu vstupnímu. Bylo to dáno tím, že 2bitová DPCM v podstatě realizovala jistou formu modulace delta a jak bylo napsáno, modulace delta potřebuje pro přenos stejného signálu jako modulace PCM/DPCM mnohem vyšší vzorkovací frekvence.

Modulace delta

Programování modulace delta bylo na rozdíl od modulací PCM a DPCM založeno na principu, kdy nebylo k přenosu využito rozhraní SPI. Výstupní signál byl rovnou nastavován na výstupní pin modulátoru. Na rozdíl od simulací, ve kterých byl výstupní signál modulátoru delta reprezentován bipolárním signálem s návratem k nule, byl při programování modulace delta použit jako výstupní signál unipolární signál bez návratu k nule. Tímto byla sice snížena přehlednost časových průběhů, ale byly tak sníženy požadavky na mikrokontrolér a program.

Adaptabilní modulace delta

Programování adaptabilní delta modulace probíhalo podle stejného principu jako programování modulace delta. Stejně jako u modulace delta byl také u adaptabilní modulace delta použit jako výstupní signál unipolární signál bez návratu k nule. To byl také zásadní rozdíl mezi simulací této modulace a měřením naprogramované modulace v laboratorních přípravcích. Stejně jako při simulacích, také při programování této modulace bylo pro adaptaci výšky schodu použito upraveného Winklerova algoritmu. Nejlepších výsledků bylo dosaženo v případě, kdy bylo nastaveno maximum výšky schodu přibližně třetinu maximální velikosti vstupního signálu a minimum tohoto parametru na přibližně dvacetinu maxima vstupního signálu. Pokud tedy byl vstupní signál v rozsahu 0V – 5V, ideální maximální hranicí výšky schodu byla hodnota 1,6V a minimální hranicí výšky schodu byla hodnota 0,2V.

Při vyšší maximální hranici výšky schodu bylo sice možné dosáhnout vyšších frekvencí vstupního signálu, aniž by docházelo k přetížení kodéru, avšak výstupní signál demodulátoru se velmi odchyloval od vstupního signálu modulátoru.

Při nižší minimální hranici výšky schodu bylo možné dosáhnout hladšího výstupního signálu demodulátoru, avšak pouze pro velmi nízké frekvence. Pokud by v tomto případě přišel na vstup modulátoru signál s vyšší dynamikou, nebyl by modulátor schopen tento signál správně modulovat.

Ze zmíněného vyplývá, že správné nastavení hranic výšky schodu je pro adaptabilní delta modulaci zásadní.

Návrh koncepce pro výrobu demonstračních přípravků

V této kapitole byla popsána koncepce pro výrobu laboratorních přípravků. Nejprve bylo zamýšleno, že každý z přípravků, modulátor i demodulátor, bude obsahovat pouze jeden mikrokontrolér. Po naprogramování prvních aplikací realizujících PCM modulaci, bylo zjištěno, že tato koncepce není ideální a proto se jako vhodnější zvolila koncepce, v níž demodulátor obsahuje jeden mikrokontrolér, stejně jako tomu bylo v předchozí koncepci. Rozdíl obou koncepcí byl v modulátoru. Nová koncepce obsahovala dva mikrokontroléry, z nichž jeden provádí pouze A/D převod a druhý provádí samotnou modulaci.

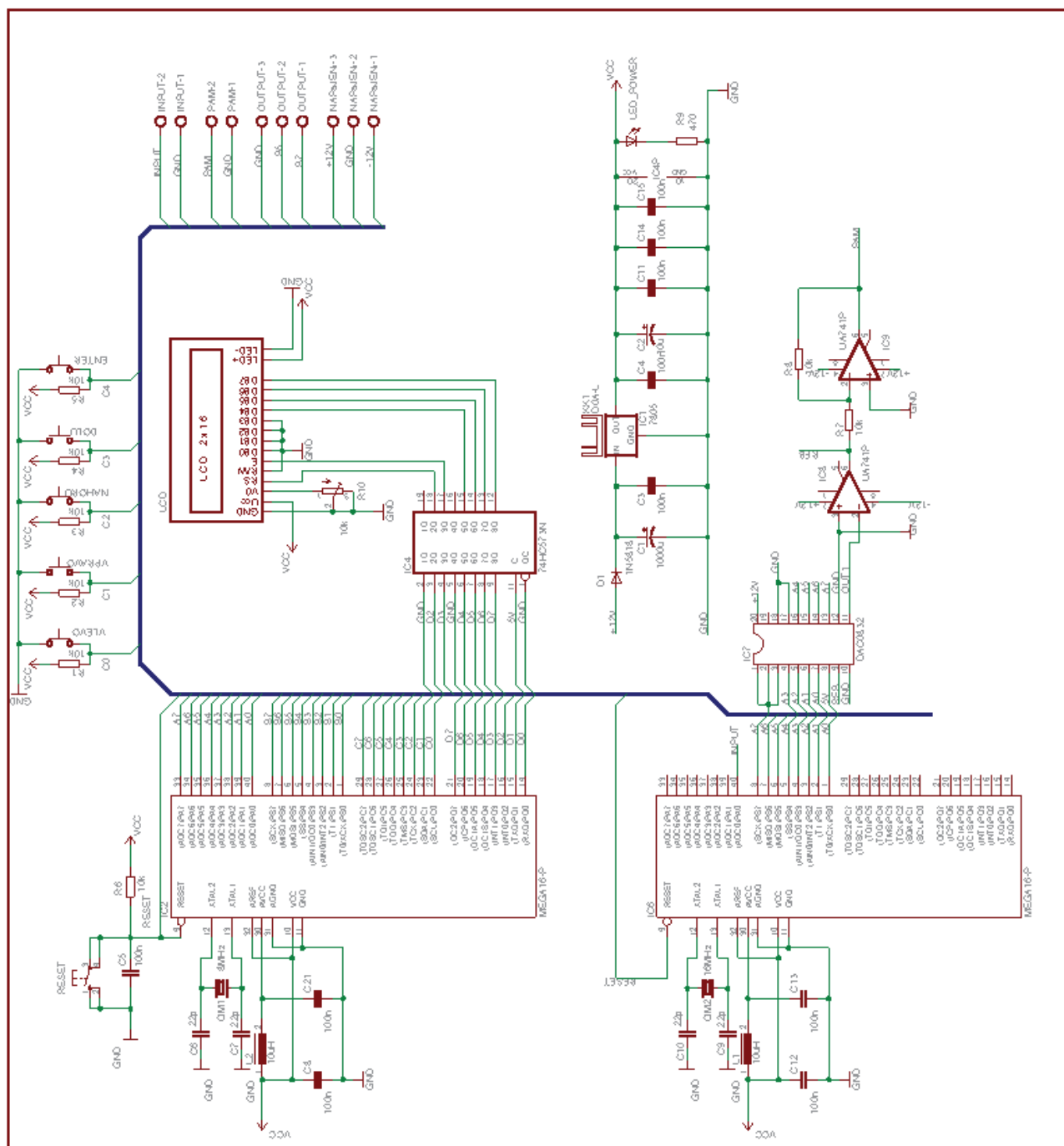
Další částí, kterou koncepce řešila, byla komunikace zařízení s uživatelem. Tato část byla vyřešena blokem komunikace, který se skládá z bloku zobrazení a bloku řízení. Blok zobrazení je tvořen displejem, na kterém uživatel může pozorovat odezvu programu na jím zadané požadavky z bloku řízení. Blok řízení je tvořen ovládacími tlačítky.

Celkové shrnutí práce

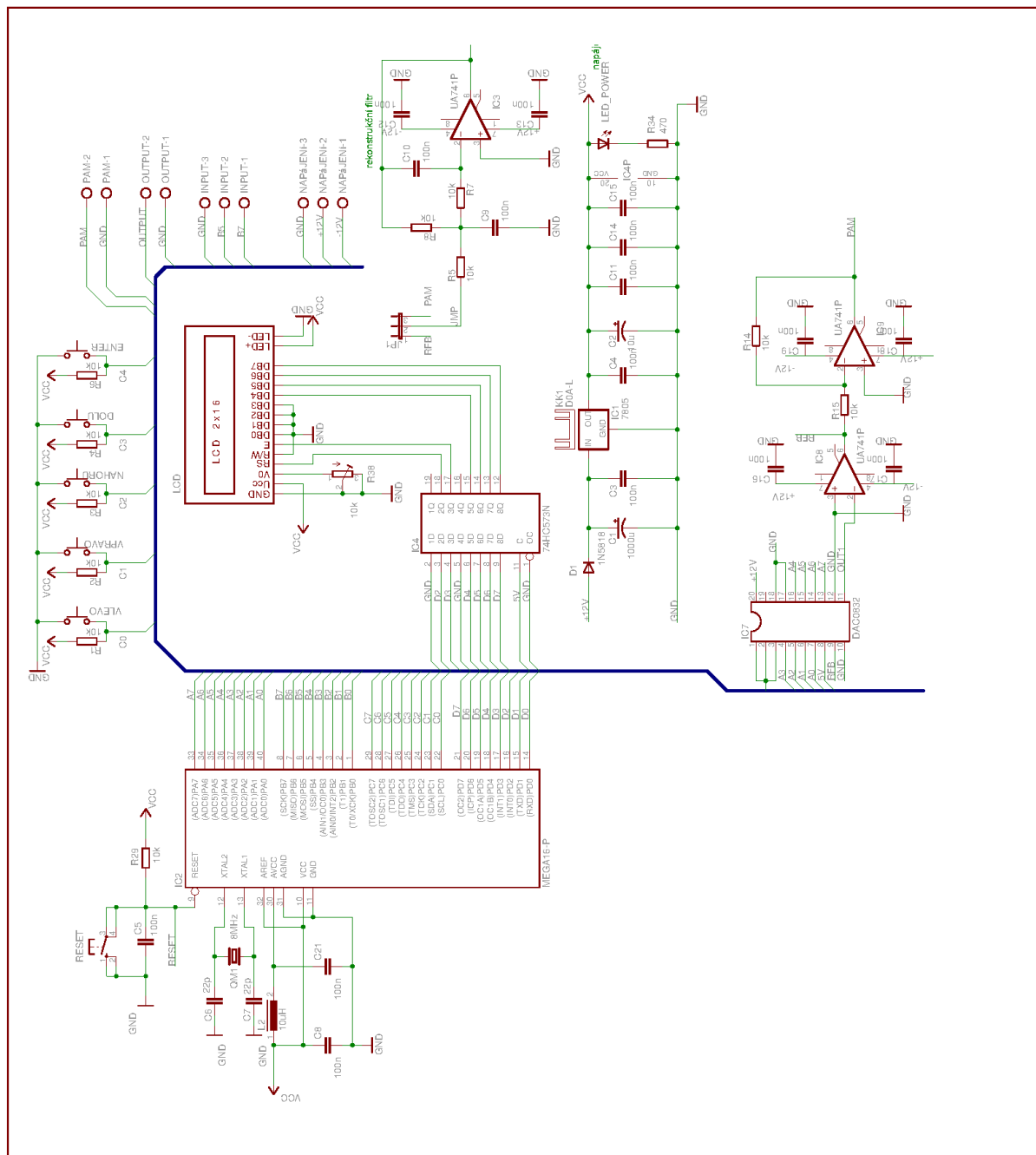
I přes některé problémy, které nastaly jak během vytváření simulačních tak během programování impulzových modulací, byly zhotoveny demonstrační modely pro všechny požadované modulace. Stejně tak byly naprogramovány všechny zvolené modulace.

Přílohy

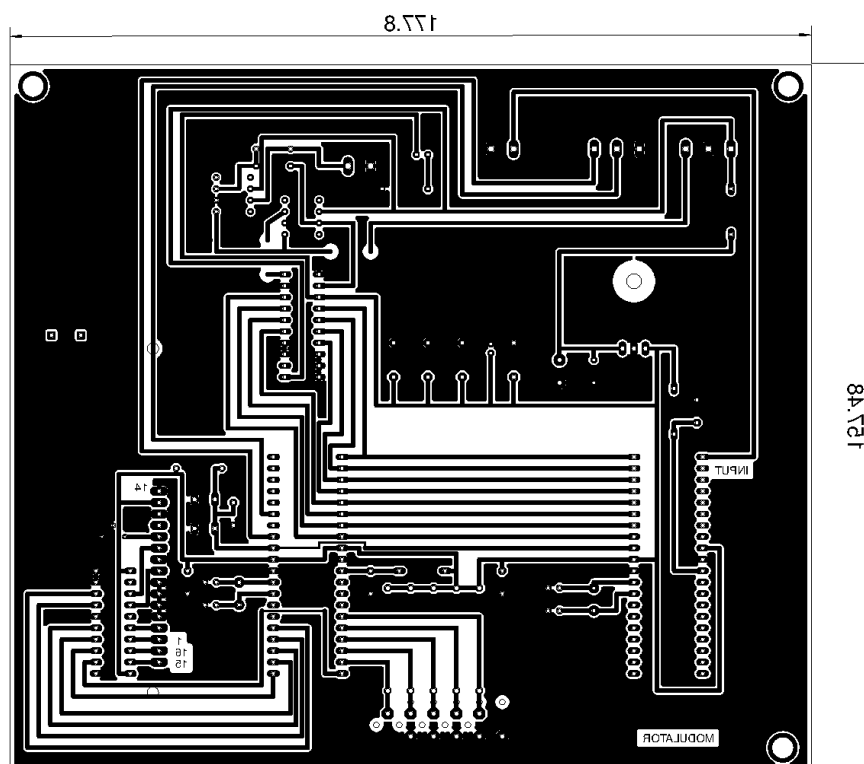
Příloha č.1: Schéma zapojení modulátoru



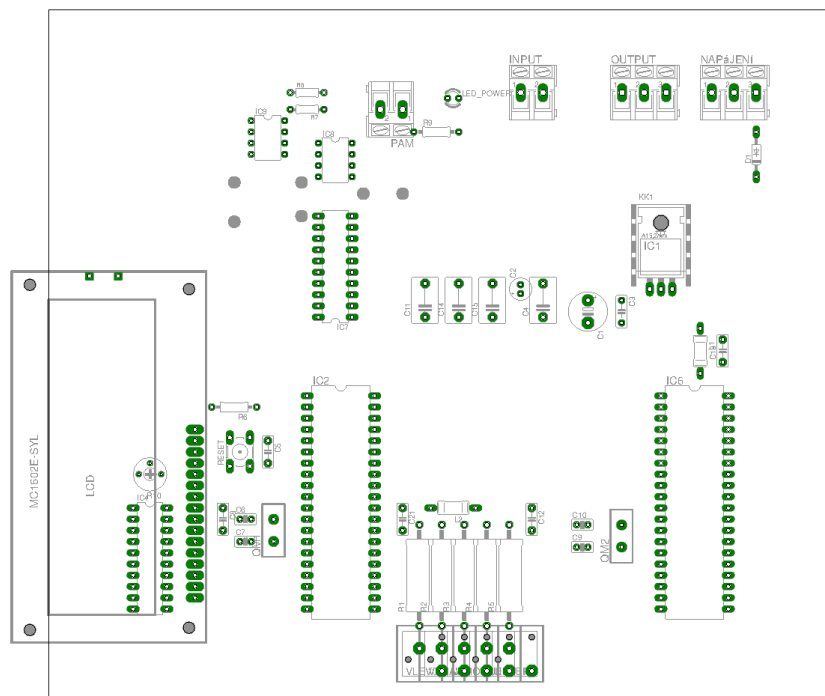
Příloha č.2: Schéma zapojení demodulátoru



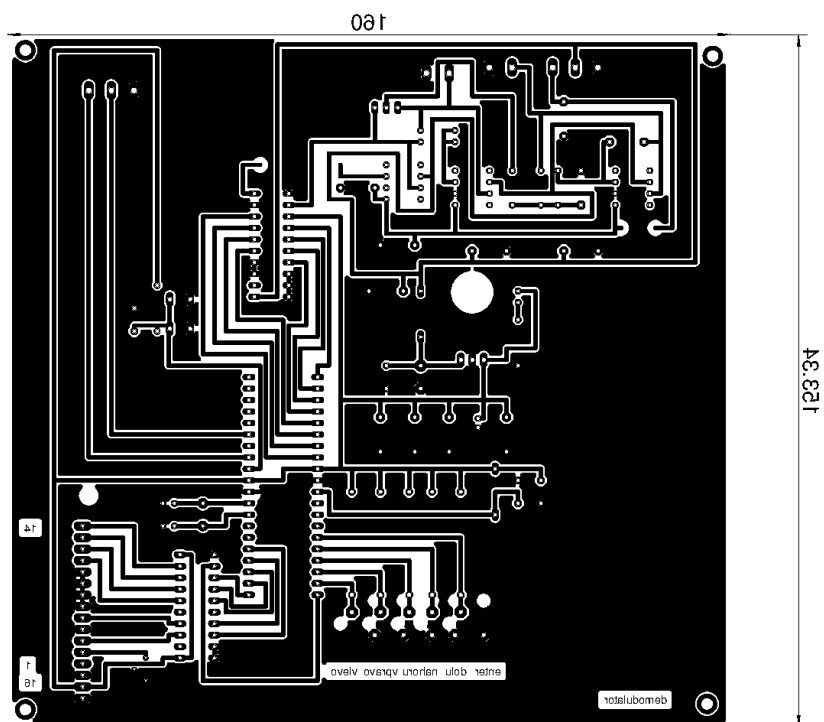
Příloha č.3: Návrh desky modulátoru



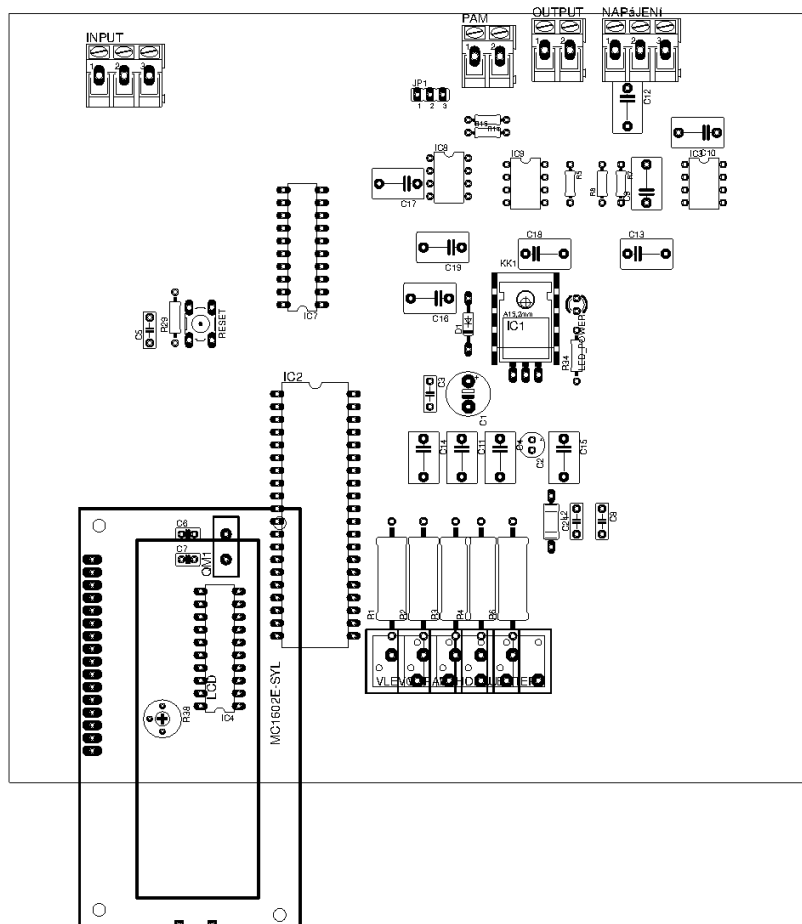
Příloha č.4: Osazovací výkres modulátoru



Příloha č.5: Návrh desky demodulátoru



Příloha č.6: Osazovací výkres demodulátoru



Příloha č.7: Seznam součástek modulátoru

Označení součástky	Hodnota	Označení součástky	Hodnota
C1	1000u	IC7	DAC0832
C2	10u	IC8	UA741P
C3	100n	IC9	UA741P
C4	100n	L1	10uH
C5	100n	L2	10uH
C6	22p	NAHORU	P-B1715
C7	22p	IC8	UA741P
C8	100n	QM1	8MHz
C9	22p	QM2	16MHz
C10	22p	R1	10k
C11	100n	R2	10k
C12	100n	R3	10k
C13	100n	R4	10k
C14	100n	R5	10k
C15	100n	R6	10k
C21	100n	R7	10k
D1	1N4148	R8	10k
DOLU	P-B1715	R9	470
ENTER	P-B1715	R10	10k
IC1	7805	RESET	P-B1715
IC2	MEGA16-P	VLEVO	P-B1715
IC4	74HC573N	VPRAVO	P-B1715
IC6	MEGA16-P	NAHORU	P-B1715
LCD	WH1602A	LED	Red LED

Příloha č.8: Seznam součástek demodulátoru

Označení součástky	Hodnota	Označení součástky	Hodnota
C1	1000u	IC4	74HC573N
C2	10u	IC7	DAC0832
C3	100n	IC8	UA741P
C4	100n	IC9	UA741P
C5	100n	LED	Red LED
C6	22p	L2	10uH
C7	22p	LCD	WH1602A
C8	100n	NAHORU	P-B1715
C9	100n	QM1	8MHz
C10	100n	R1	10k
C11	100n	R2	10k
C12	100n	R3	10k
C13	100n	R4	10k
C14	100n	R5	10k
C15	100n	R6	10k
C16	100n	R7	10k
C17	100n	R8	10k
C18	100n	R14	10k
C19	100n	R15	10k
C21	100n	R29	10k
D1	1N4148	R34	470
DOLU	P-B1715	R38	10k
ENTER	P-B1715	RESET	P-B1715
IC1	7805	VLEVO	P-B1715
IC2	MEGA16-P	VPRAVO	P-B1715
IC3	UA741P		

Literatura

- [1] BEZDĚK, Miloslav. *Elektronika I*. České Budějovice : KOPP, 2005. 271 s.
- [2] ŠEBESTA, V. *Teorie sdělování*. Brno : VUTIMUM, 1998. 92 s. ISBN 80-214-1247-X.
- [3] HÁLA, Jindřich a Jiří DOUBRAVA. *ELEKTRONIKA*. Praha: Nakladatelství dopravy a spojů, 1984.
- [4] ČÍŽ, Radim. Soubor přednášek z předmětu teorie sdělování. Dostupné z: <http://www.utko.feec.vutbr.cz/~cizr/tsd/index.php#Přednášky>
- [5] KUTĚJ, Jaroslav. Knihovna pro LCD [on-line]. Vydáno 2009 [citováno 2010-09-20]. Dostupné z: <http://mp222.wz.cz/>
- [6] NAVRÁTIL, P. Didaktická příručka pro vývojový kit s ATMEL. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 40 s., 14 s. příloh Vedoucí semestrální práce Ing. Martin Friedl.
- [7] VÁŇA, Vladimír. Mikrokontroléry ATMEL AVR: programování v jazyce C : popis a práce ve vývojovém prostředí CodeVisionAVR C. 1. vyd. Praha: BEN - technická literatura, 2003, 215 s. ISBN 80-730-0102-0
- [8] MANN, Burkhard. C pro mikrokontroléry. 1. české vydání. Praha: BEN - technická literatura, 2003, 215 s. ISBN 80-7300-077-6
- [9] ATMEL CORPORATION. ATmega16-16PI: datasheet [on-line]. 2002, 316 s. [cit. 26.5.2013]. Dostupné z: <http://www.atmel.com>
- [10] TEXAS INSTRUMENTS. DAC0832: datasheet [online]. 2002 [cit. 20.5.2013]. Dostupné z: <http://www.ti.com>
- [11] STMICROELECTRONICS. 7805-STM: datasheet [online]. 2004 [cit. 20.5.2013]. Dostupné z: <http://www.gme.cz/dokumentace/330/330-149/dsh.330-149.1.pdf>

Seznam obrázků

Obrázek č.1: Základní rozdělení impulzních modulací	8
Obrázek č.2: Zjednodušené blokové schéma modulátoru PAM	10
Obrázek č.3: PAM 1. a 2.druhu.....	10
Obrázek č.4: Zobrazení důležitých parametrů PAM II signálu	11
Obrázek č.5: Časové průběhy u modulace PPM	12
Obrázek č.6: Zjednodušené blokové schéma PWM.....	13
Obrázek č.7: Časové průběhy PWM	13
Obrázek č.8: Zjednodušené blokové schéma PCM.....	15
Obrázek č.9: Časové průběhy v modulátoru PCM.....	15
Obrázek č.10: Časový průběh nakvantovaného signálu.....	16
Obrázek č.11: Zapojení bloku kvantování	16
Obrázek č.12: Blokové schéma zapojení kodéru	17
Obrázek č.13: Výstupní signál PCM.....	18
Obrázek č.14: Časové průběhy PCM	18
Obrázek č.15: Zjednodušené blokové schéma DPCM.....	19
Obrázek č.16: Objasnění místa výskytu s maximální diferencí [4]	20
Obrázek č.17: Časové průběhy DPCM	21
Obrázek č.18: Časové průběhy pro vstupní signál s frekvencí blízkou nežní frekvenci	22
Obrázek č.19: Časové průběhy pro vstupní signál s frekvencí nižší než mezní	22
Obrázek č.20: časové průběhy pro vstupní signál s frekvencí vyšší než mezní.....	22
Obrázek č.21: Zobrazení důležitých parametrů DM	23
Obrázek č.22: Zjednodušené blokové schéma zapojení DM[4]	24
Obrázek č.23: Časové průběhy signálů v modulátoru a demodulátoru DM	25
Obrázek č.24: Zobrazení zkreslení dané přetížením kodéru	26
Obrázek č.25: Zjednodušené blokové schéma modulátoru a demodulátoru SMD [4]	27
Obrázek č.26: Časové průběhy signálů v modulátoru SDM.....	28
Obrázek č.27: Absence zkreslení dané přetížením kodéru	29
Obrázek č.28: Zjednodušené blokové schéma modulátoru a demodulátoru ADM	29
Obrázek č.29: Zjednodušené blokové schéma bloku Adaptace.....	30
Obrázek č.30: Zjednodušené blokové schéma bloku Integrátor	31
Obrázek č.31: Porovnání časových průběhů aproximačních signálů DM a ADM	31
Obrázek č.32: Časové průběhy ADM bez omezení výšky schodu	32
Obrázek č.33: Zjednodušené blokové schéma bloku Adaptace obsahující blok Saturation	33
Obrázek č.34: Porovnání časových průběhů aproximačních signálů.....	33
Obrázek č.35: Časové průběhy ADM s omezením výšky schodu	34
Obrázek č.36: Časové průběhy vstupního a výstupního signálu PCM	35
Obrázek č.37: Časové průběhy vstupního a výstupního signálu PCM	36
Obrázek č.38: Časové průběhy vstupního a výstupního signálu DPCM	37
Obrázek č.39: Časové průběhy vstupního a výstupního signálu DPCM	38
Obrázek č.40: Časové průběhy vstupního a výstupního signálu.....	39
Obrázek č.41: Propojení dvou zařízení pomocí SPI [9].....	42
Obrázek č.42: Časové průběhy signálů při přenosu přes SPI rozhraní	44
Obrázek č.43: Zjednodušené blokové schéma modulátoru PCM	45
Obrázek č.44: Časové průběhy signálů při přenosu přes SPI rozhraní	46
Obrázek č.45: Časové průběhy vstupního signálu a výstupního signálu PAM v modulátoru PCM	47
Obrázek č.46: Časové průběhy vstupního a výstupního signálu v modulátoru PCM.....	47
Obrázek č.47: Zjednodušené blokové schéma demodulátoru	48
Obrázek č.48: Časové průběhy vstupního a výstupního signálu před filtrací v demodulátoru PCM	49

Obrázek č.49: Časové průběhy výstupního signálu demodulátoru PCM před filtrací a po filtraci	49
Obrázek č.50: Časové průběhy vstupního signálu modulátoru PCM a výstupního signálu demodulátoru PCM	50
Obrázek č.51: Časové průběhy vstupního signálu a výstupního signálu PAM v modulátoru PCM	52
Obrázek č.52: Časové průběhy vstupního a výstupního signálu modulátoru PCM.....	53
Obrázek č.53: Časové průběhy vstupního signálu a nefiltrovaného výstupního signálu v demodulátoru PCM	54
Obrázek č.54: Časové průběhy výstupního signálu demodulátoru PCM před a po filtraci	55
Obrázek č.55: Časové průběhy vstupního signálu modulátoru PCM (oranžový) a výstupního signálu demodulátoru PCM (modrý).....	55
Obrázek č.56: Zjednodušené blokové schéma modulátoru DPCM	56
Obrázek č.57: Časové průběhy vstupního signálu a signálu PAM před jeho úpravou v modulátoru DPCM	57
Obrázek č.58: Časové průběhy vstupního signálu a signálu PAM po jeho úpravě v modulátoru DPCM.....	58
Obrázek č.59: Časové průběhy vstupního a výstupního signálu v modulátoru DPCM.....	58
Obrázek č.60: Časové průběhy vstupního konstantního signálu a výstupního signálu demodulátoru.....	59
Obrázek č.61: Zjednodušené blokové schéma demodulátoru DPCM.....	60
Obrázek č.62: Časové průběhy vstupního signálu a signálu reprezentující rozdíl mezi dvěma po sobě následujícími vzorky v demodulátoru DPCM	61
Obrázek č.63: Časové průběhy PAM signálu po úpravě a PAM signálu po jeho filtraci v demodulátoru DPCM	62
Obrázek č.64: Časové průběhy vstupního signálu modulátoru a výstupního signálu demodulátoru DPCM pro 500Hz a 200Hz.....	62
Obrázek č.65: Zjednodušené blokové schéma modulátoru delta modulace	63
Obrázek č.66: Časové průběhy signálů v modulátoru delta s přetížením kodéru	64
Obrázek č. 67: Časové průběhy signálů v modulátoru delta bez přetížení kodéru	64
Obrázek č.68: Časové průběhy vstupního a výstupního signálu v modulátoru delta	65
Obrázek č.69: Detail časových průběhů vstupního a výstupního signálu.....	65
Obrázek č.70: Zjednodušené blokové schéma demodulátoru delta	66
Obrázek č.71: Časové průběhy signálů v demodulátoru delta	67
Obrázek č.72: časové průběhy výstupního signálu před filtrací a po filtraci	67
Obrázek č.73: Časové průběhy vstupního signálu modulátoru a výstupního signálu demodulátoru.....	68
Obrázek č.74: Zjednodušené blokové schéma modulátoru ADM	69
Obrázek č.75: Časové průběhy vstupního signálu a úrovně rozhodovacího signálu v modulátoru ADM	70
Obrázek č.76: Časové průběhy vstupního a výstupního signálu v modulátoru ADM.....	70
Obrázek č.77: Časové průběhy úrovně rozhodovacího signálu a výstupního signálu v modulátoru ADM	71
Obrázek č.78: Zjednodušené blokové schéma demodulátoru adaptabilní delta	71
Obrázek č.79: Časové průběhy vstupního a výstupního signálu před filtrací, v demodulátoru ADM.....	73
Obrázek č.80: Časové průběhy výstupního signálu demodulátoru před filtrací a po filtraci ...	73
Obrázek č.81: Časové průběhy vstupního signálu modulátoru a výstupního signálu demodulátoru ADM	74
Obrázek č.82: Rozložení tlačítek na zařízení	74
Obrázek č.83: Výběr modulace	76
Obrázek č.84: Výběr vzorkovací frekvence	77
Obrázek č.85: Výběr počtu bitů na vzorek.....	77

Obrázek č.86: Spuštění modulace	77
Obrázek č.87: Koncepce modulátor/demodulátor.....	81
Obrázek č.88: Finální výrobky.....	83
Obrázek č.89: Pohled na zadní stranu přípravku demodulátoru	83

Seznam tabulek

Tabulka č.2.1: Důležité parametry PAM II signálu	11
Tabulka č.3.1: Nastavení výstupní hodnoty bloku Multiport Switch	30
Tabulka č.5.1: Parametry řídicí struktury.....	75
Tabulka č.6.1: Dosažené mezní frekvence pro modulaci PCM	79
Tabulka č.6.2: Dosažené mezní frekvence pro modulaci DPCM	80
Tabulka č.6.3: Dosažené vybrané mezní frekvence pro modulaci DPCM	80
Tabulka č.9.1: Maximální hodnoty kmitočtu vstupního signálu při daném vzorkovací kmitočtu u PCM	85
Tabulka č.9.2: Počet kvantizačních úrovní pro daný počet bitů u PCM.....	85
Tabulka č.9.3: Velikost kvantizačního kroku pro daný počet bitů u PCM a vstupní signál s rozsahem 5V	85
Tabulka č.9.4: Doba trvání synchronizačního pulzu pro nbitovou PCM a danou vzorkovací frekvenci.....	86
Tabulka č.9.5: Mezní frekvence vstupního signálu pro danou vzorkovací frekvenci, v případě snížení počtu bitů DPCM o 1 bit vůči počtu bitů PCM	86
Tabulka č.9.6: Mezní frekvence vstupního signálu pro danou vzorkovací frekvenci, danou výšku schodu	86

Seznam použitých zkratek

Seznam použitých zkratek

Zkratka použitá v textu	Význam použité zkratky
A	Amplituda signálu
ADEN	Registr k povolení A/D převodu
ADIF	Registr signalizující dokončení A/D převodu
ADM	Adaptabilní delta modulace
ADMUX	Registr pro nastavení A/D převodníku
ADCSRA	Řídící registr A/R převodu v ATmega16
ADSC	Bit započínající A/D převod
AM	Amplitudová modulace
A/D	Analogově - digitální (převodník)
CLK	Hodinový signál
DATA	Označení datového vstupu/výstupu
DM	Delta modulace
DPCM	Diferenční pulzně kódová modulace
D/A	Digitálně – analogový (převodník)
FM	Frekvenční modulace
$f_{\max}$	Maximální frekvence
f_{OSC}	Frekvence oscilátoru
FS	Full scale = plný rozsah
f_{vz}	Vzorkovací frekvence
LCD	Displej z tekutých krystalů
MSTR	Registr pro nastavení zařízení do módu MASTER
N_{KV}	Počet kvantovacích úrovní
PAM	Pulzně amplitudová modulace
PCM	Pulzně kódová modulace
PM	Fázová modulace
PPM	Pulzně polohová modulace
PWM	Pulzně šířková modulace
REFS	Bit pro výběr referenčního napětí
SDM	Sigma delta modulace
SPCR	Řídící registr SPI v ATmega16
SPE	Registr povolující SPI přenos (SPI enabled)
SPI	Sériové rozhraní ATmega16
SPIF	Registr signalizující konec SPI přenosu
SPR	Bit k nastavení předděličky frekvence u SPI
S-H	Sample and Hold vzorkování
T_{vz}	Vzorkovací perioda