



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNologiÍ**  
**ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ**

**FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION**  
**DEPARTMENT OF BIOMEDICAL ENGINEERING**

## **OSCILOSKOP V SYSTÉMU ANDROID**

**OSCILLOSCOPE WITH THE ANDROID OPERATING SYSTEM**

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**RICHARD PAŠTĚKA**

**VEDOUcí PRÁCE**  
SUPERVISOR

**Ing. JIŘÍ SEKORA**

**BRNO 2014**



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav biomedicínského inženýrství

# Bakalářská práce

bakalářský studijní obor  
**Biomedicínská technika a bioinformatika**

**Student:** Richard Paštěka

**ID:** 134379

**Ročník:** 3

**Akademický rok:** 2013/2014

**NÁZEV TÉMATU:**

## Osciloskop v systému Android

### POKYNY PRO VYPRACOVÁNÍ:

1) Seznamte se s laboratorním osciloskopem a s jeho možnými funkcemi. Seznamte se s vývojovým kitem Arduino pro systém Android a se způsobem programování systému Android. 2) S využitím hardwarového kitu Arduino navrhnete jednoduchý osciloskop alespoň těchto parametrů: snímání střídavého i stejnosměrného napětí v rozsahu -5 V až +5 V. Maximální frekvenci osciloskopu stanovte dle možností vývojového kitu, možnost sejmutí a uložení aktuální obrazovky, možnost ruční i automatické změny časového i napěťového měřítka. Přesnost AD převodníku a tím i osciloskopu bude dána rozsahem ADC vývojového kitu. Vzorkovací frekvence bude dána možnostmi vývojového kitu. 3) Sestavte grafické uživatelské rozhraní pro platformu Android. 4) Návrh osciloskopu realizujte a otestujte pomocí funkčního generátoru.

### DOPORUČENÁ LITERATURA:

[1] OXER, Jonathan. Practical Arduino: cool projects for open source hardware. Berkeley, CA: Apress, c2009, xx, 423 p. ISBN 978-1-4302-2477-8.

[2] UJBÁNYAI, Miroslav. Programujeme pro Android. Vyd. 1. Praha: Grada, 2012, 187 s. Průvodce (Grada). ISBN 978-80-247-3995-3.

**Termín zadání:** 10.2.2014

**Termín odevzdání:** 30.5.2014

**Vedoucí práce:** Ing. Jiří Sekora

**Konzultanti bakalářské práce:**

**prof. Ing. Ivo Provazník, Ph.D.**

*Předseda oborové rady*

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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

Bakalářská práce se zabývá návrhem digitálního osciloskopu pomocí vývojové platformy Arduino Mega ADK. V teoretické části práce jsou rozebrány základní parametry digitálních osciloskopů, typy vzorkování a obecné vlastnosti A/D převodníků. Práce se dále zabývá popisem vývojové platformy Arduino Mega ADK. Praktická část práce popisuje jednotlivé kroky realizace osciloskopu. Od obecného návrhu přechází v konkrétní popis řešení softwaru a hardwaru. Software se skládá ze dvou navzájem komunikujících programů. První, program mikrokontroléru, obstarává akvizici dat. Druhý slouží k vykreslení měřených průběhů na obrazovku počítače. Hardware přecházející platformě Arduino Mega ADK vhodně upravuje vstupní měřený signál pro účely měření. Ve výsledcích jsou zhodnoceny průběhy a parametry vytvořeného osciloskopu, které byly otestovány pomocí funkčního generátoru.

## KLÍČOVÁ SLOVA

digitální osciloskop, Arduino ADK, Matlab, A/D převodník s postupnou aproximací

## ABSTRACT

Bachelor thesis deals with the design of digital oscilloscope using the Arduino development kit. The theoretical part of the thesis discusses the basic parameters of digital oscilloscopes, sampling types and general characteristics of A/D converters. The thesis further focuses on the description of the development platform Arduino Mega ADK. The practical part describes the steps in the implementation of the oscilloscope. The general design transfers to a specific description of software and hardware solutions. The software consists of two mutually communicating programs. First, program of the microcontroller performs data acquisition. The second is used to render the measured waveforms on the computer screen. Hardware appropriately adjusts the measuring signal for measurement purposes before Arduino Mega ADK platform. In the results the evaluation of the created oscilloscope waveforms and parameters is given tested by using a function generator.

## KEYWORDS

digital oscilloscope, Arduino ADK, Matlab, successive approximation ADC

PAŠTĚKA, Richard *Osciloskop v systému Android*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2014. 67 s. Vedoucí práce byl Ing. Jiří Sekora,

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Osciloskop v systému Android“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Jiřímu Sekorovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

(podpis autora)

# OBSAH

<b>Úvod</b>	<b>11</b>
<b>1 Číslicové Osciloskopy</b>	<b>12</b>
1.1 Vlastnosti číslicových osciloskopů . . . . .	13
1.2 Vzorkování signálu . . . . .	14
1.2.1 Typy vzorkování . . . . .	16
1.2.2 Chyby vzorkování . . . . .	18
1.3 Digitalizace signálu . . . . .	20
1.3.1 Obecné vlastnosti A/D převodníků . . . . .	20
1.3.2 A/D převodníky s postupnou aproximací . . . . .	23
1.3.3 Chyby A/D převodníků . . . . .	25
<b>2 Platforma Arduino</b>	<b>28</b>
2.1 Hardwarová část platformy Arduino . . . . .	28
2.2 Softwarová část platformy Arduino . . . . .	28
2.3 Arduino Mega ADK . . . . .	29
2.3.1 Vstupní a výstupní napětí . . . . .	29
2.3.2 Paměť . . . . .	30
2.3.3 Komunikace . . . . .	30
2.3.4 Nadproudová ochrana sběrnice USB . . . . .	30
<b>3 Realizace osciloskopu</b>	<b>31</b>
3.1 Návrh osciloskopu . . . . .	31
3.2 Software osciloskopu . . . . .	32
3.2.1 Program akviziciční části . . . . .	33
3.2.2 Aplikace zobrazení . . . . .	35
3.2.3 Komunikace . . . . .	38
3.3 Hardware osciloskopu . . . . .	39
3.3.1 Zesílení vstupního signálu . . . . .	39
3.3.2 Offset vstupního signálu . . . . .	41
3.3.3 Koncový stupeň . . . . .	43
3.3.4 Napájení . . . . .	43
3.3.5 Proudový odběr . . . . .	45
<b>4 Naměřené parametry a průběhy</b>	<b>46</b>
4.1 Rozsah vstupního napětí . . . . .	46
4.2 Počet kanálů . . . . .	46
4.3 Vzorkovací frekvence . . . . .	46

4.4	Horní mezní frekvence . . . . .	49
4.5	Naměřené průběhy . . . . .	49
4.5.1	Sinusový průběh . . . . .	49
4.5.2	Trojúhelníkový průběh . . . . .	51
4.5.3	Obdelníkový průběh . . . . .	52
4.5.4	Měření zesílení vstupního signálu . . . . .	54
4.5.5	Měření rozsahu vstupního signálu . . . . .	57
4.5.6	Zhodnocení naměřených výsledků . . . . .	59
<b>Závěr</b>		<b>61</b>
<b>Literatura</b>		<b>62</b>
<b>Seznam symbolů, veličin a zkratk</b>		<b>64</b>
<b>Seznam příloh</b>		<b>65</b>
<b>A Kompletní schéma zapojení</b>		<b>66</b>
<b>B Seznam použitých součástek</b>		<b>67</b>

# SEZNAM OBRÁZKŮ

1.1	Displej osciloskopu . . . . .	12
1.2	Blokové schéma n-kanálového osciloskopu . . . . .	13
1.3	Vznik záznejového signálu . . . . .	15
1.4	Jednoduchý vzorkovací zesilovač . . . . .	16
1.5	Vzorkování v reálném čase . . . . .	17
1.6	Sekvenční ekvivalentní vzorkování . . . . .	17
1.7	Náhodné ekvivalentní vzorkování . . . . .	18
1.8	Princip vzorkování v čase . . . . .	19
1.9	Digitalizace spojitého signálu . . . . .	20
1.10	Převodní charakteristika 8bitového A/D převodníku . . . . .	21
1.11	Rozdělení ADC podle rozlišení v závislosti na četnosti převodu . . . . .	22
1.12	Kompenzační ADC s postupnou aproximací . . . . .	24
1.13	Převod s postupnou aproximací, časový diagram pro a) velké $u_{VST}$ b) malé $u_{VST}$ . . . . .	25
1.14	Chyba zesílení A/D převodníku . . . . .	25
1.15	Chyba nuly A/D převodníku . . . . .	26
1.16	Kvantovací chyba ideálního A/D převodníku . . . . .	26
3.1	Blokové schéma navrhovaného osciloskopu . . . . .	31
3.2	Vývojový diagram akviziční části . . . . .	34
3.3	Vývojový diagram aplikace zobrazení . . . . .	36
3.4	Grafické uživatelské rozhraní . . . . .	37
3.5	Zapojení vstupního členu - napěťový sledovač . . . . .	40
3.6	Zapojení neinvertujícího zesilovače . . . . .	40
3.7	Zapojení sumačního OZ . . . . .	41
3.8	Blokové schéma obvodu napájení . . . . .	44
4.1	Blokové schéma generování vzorkovací frekvence A/D převodníku . . . . .	47
4.2	Sinusový průběh z funkčního generátoru $f = 500 \text{ Hz}$ , $U_{PK-PK} = 10 \text{ V}$ . . . . .	49
4.3	Sinusový průběh za koncovým stupněm $f = 500 \text{ Hz}$ , $U_{PK-PK} = 5,04 \text{ V}$ . . . . .	50
4.4	Sinusový průběh naměřený vytvořeným osciloskopem $f = 500 \text{ Hz}$ , $U_{PK-PK} = 10 \text{ V}$ . . . . .	50
4.5	Trojúhelníkový průběh z funkčního generátoru $f = 500 \text{ Hz}$ , $U_{PK-PK} = 10 \text{ V}$ . . . . .	51
4.6	Trojúhelníkový průběh naměřený za koncovým stupněm $f = 500 \text{ Hz}$ , $U_{PK-PK} = 5,04 \text{ V}$ . . . . .	51
4.7	Trojúhelníkový průběh naměřený vytvořeným osciloskopem $f = 500$ $\text{Hz}$ , $U_{PK-PK} = 10 \text{ V}$ . . . . .	52



4.8	Obdelníkový průběh z funkčního generátoru $f = 500 \text{ Hz}$ , $U_{PK-PK} = 10 \text{ V}$ . . . . .	52
4.9	Obdelníkový průběh naměřený za koncovým stupněm $f = 500 \text{ Hz}$ , $U_{PK-PK} = 5 \text{ V}$ . . . . .	53
4.10	Obdelníkový průběh naměřený vytvořeným osciloskopem $f = 500 \text{ Hz}$ , $U_{PK-PK} = 10 \text{ V}$ . . . . .	53
4.11	Sinusový průběh z funkčního generátoru $f = 500 \text{ Hz}$ , $U_{PK-PK} = 1 \text{ V}$	54
4.12	Zesílený sinusový průběh naměřený za koncovým stupněm $f = 500 \text{ Hz}$ , $U_{PK-PK} = 5 \text{ V}$ . . . . .	54
4.13	Zesílený sinusový průběh naměřený vytvořeným osciloskopem $f = 500 \text{ Hz}$ , $U_{PK-PK} = 10 \text{ V}$ . . . . .	55
4.14	Sinusový průběh po průchodu odporovým děličem $f = 500 \text{ Hz}$ , $U_{PK-PK} = 100 \text{ mV}$ . . . . .	55
4.15	Sinusový průběh naměřený za koncovým stupněm $f = 500 \text{ Hz}$ , $U_{PK-PK} = 100 \text{ mV}$ . . . . .	56
4.16	Sinusový průběh naměřený vytvořeným osciloskopem $f = 500 \text{ Hz}$ , $U_{PK-PK} = 100 \text{ mV}$ . . . . .	56
4.17	Sinusový průběh z funkčního generátoru $f = 500 \text{ Hz}$ , $U_{PK-PK} = 18,8 \text{ V}$	57
4.18	Sinusový průběh naměřený za koncovým stupněm $f = 500 \text{ Hz}$ , $U_{PK-PK} = 5 \text{ V}$ . . . . .	58
4.19	Sinusový průběh naměřený vytvořeným osciloskopem $f = 500 \text{ Hz}$ , $U_{PK-PK} = 5 \text{ V}$ . . . . .	58
4.20	Sinusový průběh naměřený za koncovým stupněm $f = 4000 \text{ Hz}$ , $U_{PK-PK} = 5 \text{ V}$ . . . . .	59
4.21	Sinusový průběh naměřený vytvořeným osciloskopem $f = 4000 \text{ Hz}$ , $U_{PK-PK} = 5 \text{ V}$ . . . . .	60

# SEZNAM TABULEK

1.1	Základní typy převodníků AD - výhody,nevýhody . . . . .	23
4.1	Nastavení hodnoty dělicího poměru . . . . .	48

# ÚVOD

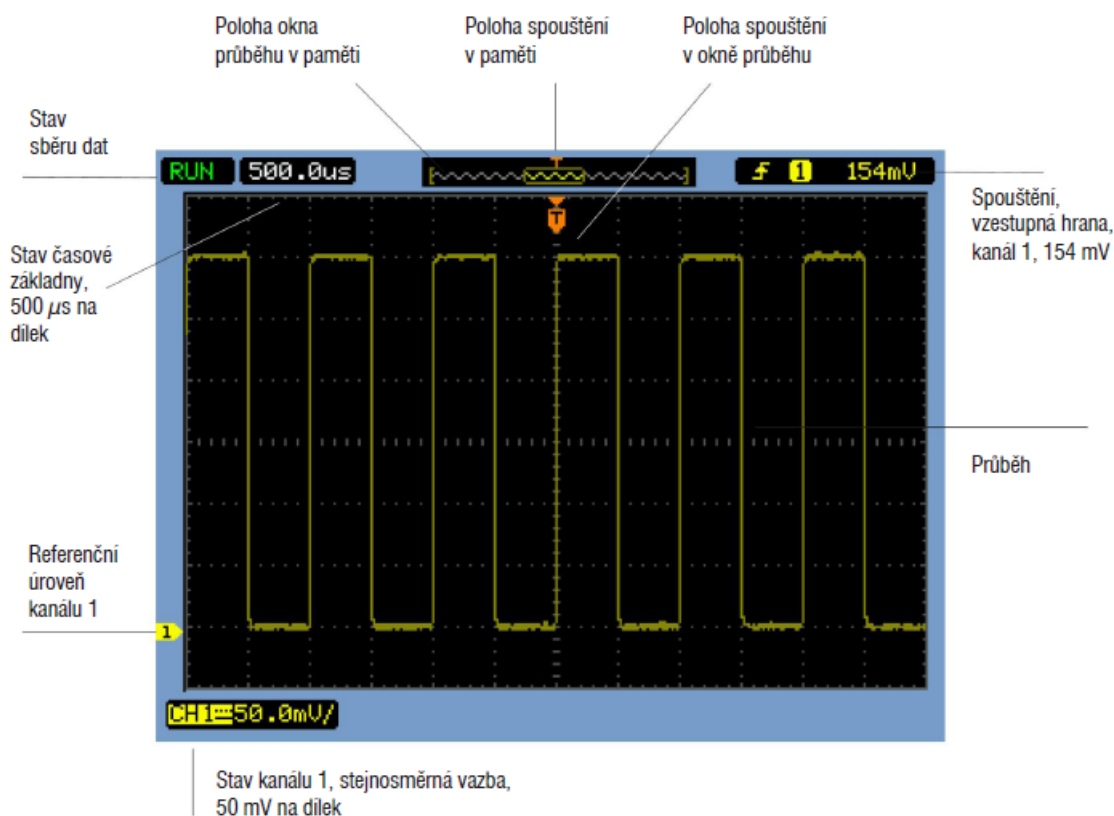
Nedílnou součástí procesu výroby, oživování, servisu a studia elektroniky je dnes bezesporu měřicí technika. Společně se vzrůstající složitostí vyráběné elektroniky se taktéž zvyšují nároky na přístroje měřicí. Osciloskopy patří k často užívaným přístrojům ve slaboproudé elektrotechnice. Jejich hlavní výhodou je, že poskytují přehlednou, rychlou a obsažnou informaci o dějích v měřených obvodech.

Tato bakalářská práce se zabývá tvorbou osciloskopu realizovaného pomocí vývojového kitu Arduino Mega ADK. Jedná se původně o prototypovací platformu pro studenty. Projekt vznikl v Itálii v roce 2005 a zaznamenal velký rozvoj a úspěch po celém světě.

V teoretické části bakalářské práce jsou popsány základní vlastnosti osciloskopů spolu s jejich nejvýznamějšími parametry, způsoby vzorkování a digitalizace pomocí A/D převodníků. Shrnuty jsou taktéž chyby, které těmito procesy vznikají. Popis platformy Arduino zahrnuje její nejdůležitější součásti. V praktické části bakalářské práce je potom představen navržený osciloskop a realizováno jeho softwarové i hardwarové vybavení. Limitace návrhu vychází především z omezených možností vývojového kitu Arduino Mega ADK. V závěrečné části je vytvořený osciloskop otestován pomocí funkčního generátoru a naměřené výsledky zhodnoceny.

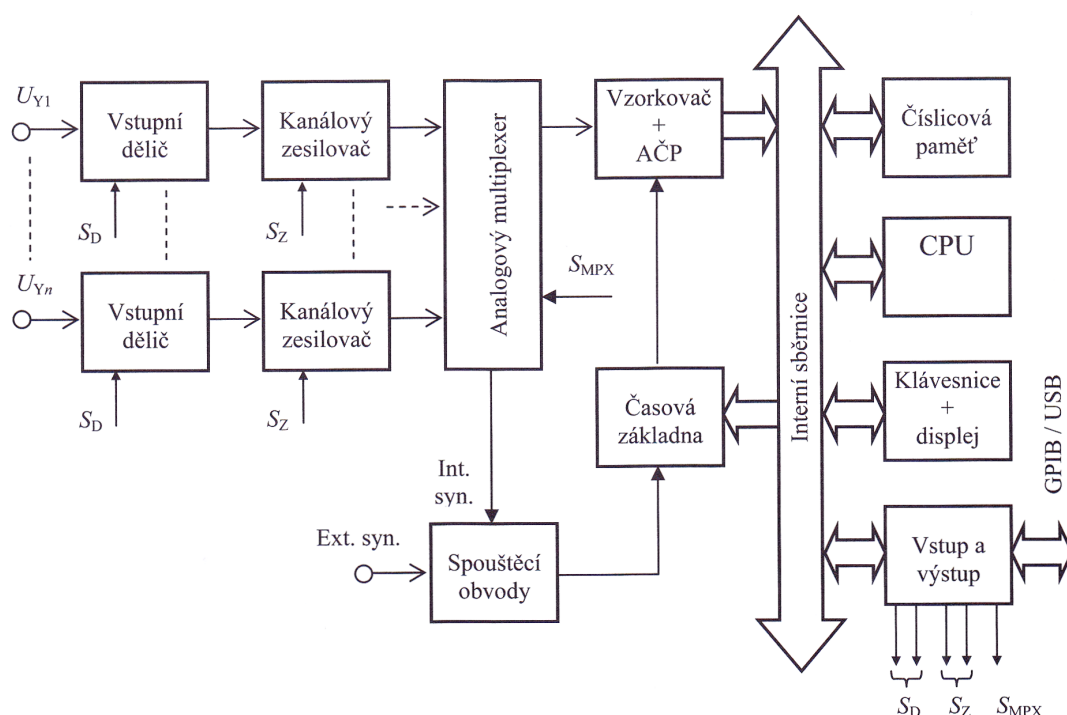
# 1 ČÍSLICOVÉ OSCILOSKOPY

Číslicové osciloskopy jsou přístroje určené k digitalizaci, rekonstrukci a zobrazení spojitých signálů. Pro své technické parametry dnes již nahradily osciloskopy analogové. V následujícím výčtu jsou uvedeny hlavní výhody, které číslicové osciloskopy nabízejí. Jedná se především o: možnost zobrazení jednorázových přechodných dějů, záznam souběžně z více kanálů i před příchodem spouštěcího impulsu, velká přesnost, automatické nastavení měřítek os, uložení průběhů do paměti a jejich číslicové zpracování, automatický výpočet a zobrazení parametrů měřeného signálu na LCD monitor, snadná archivace naměřených dat pro postprocessing (pozdější analýzu). Samozřejmostí je již dnes komunikace s počítači a jinými zařízeními prostřednictvím standardizovaných rozhraní (RS-232, USB, GPIB, IEEE 488 atd.). Displej osciloskopu společně s jeho popisem je uveden na obr. 1.1. Funkce osciloskopů může být definována jako zobrazení časového průběhu napětí ( $Y \rightarrow t$ ) nebo závislost jednoho napětí na druhém ( $Y \rightarrow X$ ).[3]



Obr. 1.1: Displej osciloskopu [2]

Na druhé straně si musíme být vědomi toho, že digitální osciloskopy pracují na principu vzorkování vstupního signálu. Uplatní se tedy jevy, které vzorkování signálu doprovází (aliasing, omezená přesnost A/D převodníku). Co největší omezení těchto nevýhod je jedním z trendů, kterým se současný vývoj digitálních osciloskopů ubírá. Mezi další negativní vlastnosti patří složitější obsluha. Při špatné manipulaci s osciloskopem můžeme dojít k zavádějícím výsledkům. Jen výjimečně je možná modulace jasem. Blokové schéma n-kanálového osciloskopu je uvedeno na obr. 1.2.[3] [9]



Obr. 1.2: Blokové schéma n-kanálového osciloskopu [7]

## 1.1 Vlastnosti číslicových osciloskopů

Mezi základní vlastnosti číslicových osciloskopů patří:

- **Počet kanálů** je dán počtem současně zobrazitelných průběhů. Běžně se jedná o 2 až 4 kanály, speciální varianty mohou mít kanálů více.[7]
- **Horní mezní frekvence** definuje pásmo harmonických signálů, které mohou být osciloskopem zobrazeny. Prakticky je horní mezní frekvence omezena vstupními obvody, zejména pak rychlostí A/D převodníku. Její maximální hodnota je omezena vzorkovacím teorémem. Více kapitola 1.2.[3]

- **Vstupní napěťové rozsahy** jsou dány napěťovým rozsahem vertikálního kanálu  $U_r$  a počtem bitů  $n$  A/D převodníku

$$\Delta U = \frac{U_r}{2^n} \quad (1.1)$$

Většina číslicových osciloskopů má napěťové rozsahy 1 mV/dílek až 5 V/dílek. Např. osciloskop s 8bitovým A/D převodníkem má na rozsahu 1 V/dílek rozlišitelnost  $\Delta U = 39$  mV.[3]

- **Vstupní impedance** je typicky dána paralelní kombinací rezistoru 1 M $\Omega$  a kapacity 5 pF - 50 pF. Při měření vyšších frekvencí bývá možnost přepnutí na 50  $\Omega$  kvůli přizpůsobení koaxiálního vedení.
- **Přesnost časové základny** (*accuracy of time base*) je určena nestabilitou vzorkovací frekvence a vzorkovacího obvodu. Dosahuje hodnot 0,1 % až 0,01 %.[3]
- **Rozlišovací schopnost A/D převodníku** (*resolution*) se vyjadřuje počtem bitů A/D převodníku. Počet bitů nejčastěji bývá 8.[3]
- **Vzorkovací frekvence** (*sampling frequency*) je nejsledovanější údaj a patří mezi základní kritéria výběru osciloskopu. Uvádí se v milionech nebo miliardách vzorků za sekundu (MS/s, *megasamples per second* GS/s, *gigasamples per second*). Současné digitální osciloskopy pracují nejčastěji v rozmezí od 100 M/s do desítek GS/s.[15]
- **Šířka pásma** (*bandwidth*) se uvádí v Hz pro pokles přenosu signálu o 3 dB vůči ss. přenosu. Zpravidla je určena antialiasingovou dolní propustí.[7]
- **Velikost paměti na kanál** (*memory per channel*) paměť je společná a distribuovaná mezi jednotlivé kanály. Jedná se o jeden z hlavních parametrů určující cenu osciloskopu. Velikost paměti se běžně pohybuje v rozmezí od desítek kB až po stovky MB. Velká kapacita paměti umožňuje vzorkovat delší časový interval nebo vzorkovat hustěji.[3]
- **Časová nestabilita spouštění** (*jitter*) určuje nejistotu spouštěcích obvodů. U rychlých signálů se projevuje „chvěním“ průběhu na obrazovce. To je způsobeno tím, že okamžik spuštění časové základny definovaný např. spouštěcí úrovní v čase kolísá kolem střední hodnoty.[7]

## 1.2 Vzorkování signálu

Vzorkováním signálu rozumíme odběr vzorku vstupního proměnného signálu, čili hodnoty signálu, v pravidelných časových intervalech s periodou  $T_v$ . Jedná se tedy o převod signálu v čase spojitým na nespojitý. Tuto diskretizaci signálu v čase provádí vzorkovač (*sampler*). Schéma jednoduchého vzorkovače je na obr. 1.4.[7][15]

$$x(t) \rightarrow x[n \cdot T_V] \quad (1.2)$$

kde  $n$  je celé číslo, značící pořadí vzorku (-)

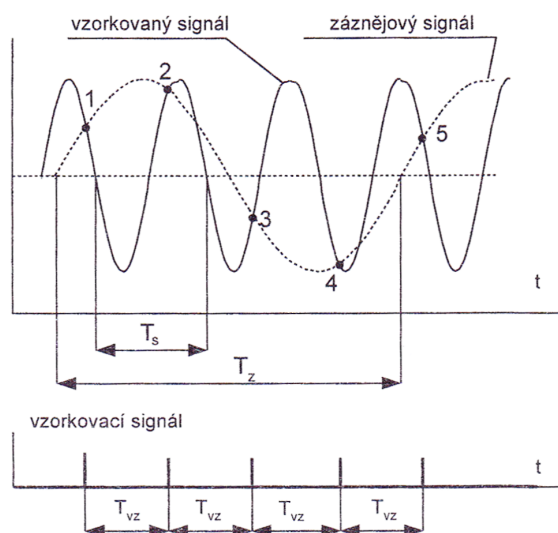
Aby mohl být vzorkovaný průběh beze ztráty informace rekonstruován na průběh původní, je nutné splnit (Nyquistův) vzorkovací teorém. Ten určuje minimální potřebnou vzorkovací frekvenci  $f_{vz}$ , která musí být rovna více než dvojnásobku nejvyššího uvažovaného kmitočtu spektra vzorkovaného signálu  $f_{max}$ [9]

$$f_{vz} > 2 \cdot f_{max} \quad (1.3)$$

Vzorkování má za následek vytvoření periodického frekvenčního spektra s periodou odpovídající  $f_{vz}$ . Není-li vzorkovací teorém splněn, dochází k překrytí sousedních period spektra (aliasing). Při rekonstrukci signálu poté vzniká záznějový signál, obr. 1.3 s frekvencí danou v nejjednodušším případě vztahem

$$f_z = |f_s - f_{vz}| \quad (1.4)$$

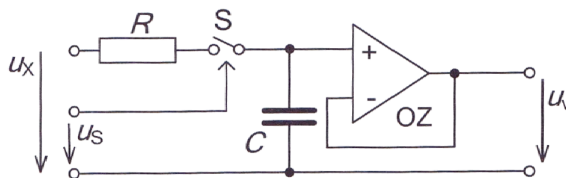
Proto je na vstup každého číslicového osciloskopu před vzorkovač zařazen dolno-propustný filtr (antialiasing-filtr) s mezní frekvencí  $f_{vz}/2$ . Na antialiasingový filtr jsou kladeny vysoké nároky. Zejména požadujeme co největší strmost a lineární fázovou charakteristiku. Pro tyto vysoké nároky se běžně používají filtry osmého až desátého řádu (Besselova či Butterworthova aproximace).[3] [7]



Obr. 1.3: Vznik záznějového signálu [7]

Dále se používají *vzorkovače s pamětí*, které se často označují jako obvod S/H *sample-and-hold*. Umožňují udržet vzorek signálu v analogové paměti po určitou dobu,

potřebnou k dalšímu zpracování A/D převodníkem. Paměťovým prvkem je v tomto obvodu kondenzátor  $C$ , který je přes spínač  $S$  řízený periodickým signálem  $u_S$  připojován na vstup vzorkovače  $u_X$ . Vybíjení kondenzátoru  $C$  do obvodu připojeného na výstup vzorkovače brání operační zesilovač OZ. V reálných obvodech se k dosažení ideálních vlastností vzorkovače používají složitá zpětnovazební zapojení s OZ.[9]



Obr. 1.4: Jednoduchý vzorkovací zesilovač [3]

Podobně jako vzorkovač s pamětí pracuje *sledovač s pamětí*, který se často označuje (T/H, *track-hold*). V tomto případě se kondenzátor  $C$  nabíjí až na hodnotu napětí shodnou s napětím vstupní. Doba nabíjení nezávisí na délce řídicího impulsu. Zároveň je v porovnání s obvodem S/H podstatně delší délka čtení vstupu. Doba pamatování využitelná k digitalizaci se zkracuje a je tedy nutno použít rychlejší A/D převodníky. [7]

### 1.2.1 Typy vzorkování

Pro vzorkování spojitého vstupního signálu se v číslicových osciloskopech používají tři typy vzorkování:

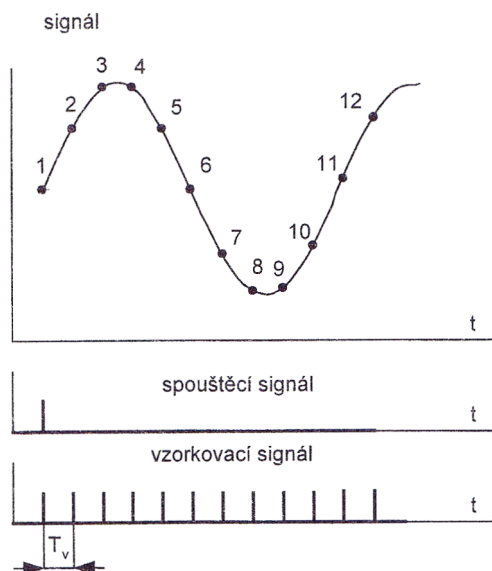
1. **Vzorkování v reálném čase** (*real-time sampling*) se uskutečňují v časově ekvidistantních vzorkovacích intervalech  $T_V$ , které určují vzorkovací frekvenci  $f_{vz}$  v reálném čase (real time sample rate). Příklad vzorkování v reálném čase je uveden na obr. 1.5 Vzorkovací frekvence musí plnit podmínku 1.3 , přičemž v praxi bývá mnohonásobně vyšší. Vzorkování v reálném čase umožňuje užít metodu zobrazení signálu před příchodem spouštěcího signálu, tzv. záporné zpoždění (*Pretriggering*).[3]
2. **Sekvenční vzorkování v ekvivalentním čase** (*sequential repetitive sampling*) je charakteristické odběrem vzorků z *periodického signálu* tak, že se postupně zvětšuje doba mezi jednotlivými odběry o konstantní přírůstek  $\Delta T$ , obr. 1.6. Z takto odebraného velkého počtu vzorků, daného kapacitou paměti, je rekonstruován původní průběh signálu. Z každé periody je odebrán jeden vzorek. Odběr je inicializován spouštěcím impulzem. Ekvivalentní vzorkovací



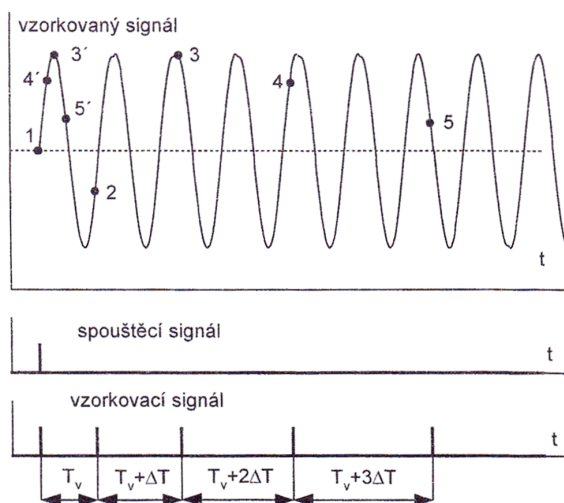
frekvence je dána vztahem:

$$f_{vz} = \frac{1}{\Delta T} \quad (1.5)$$

Proto je možno zachytit více detailů než při vzorkování v reálném čase. Předpokladem je však zaručení periodicity vzorkovaného signálu.[7][3]

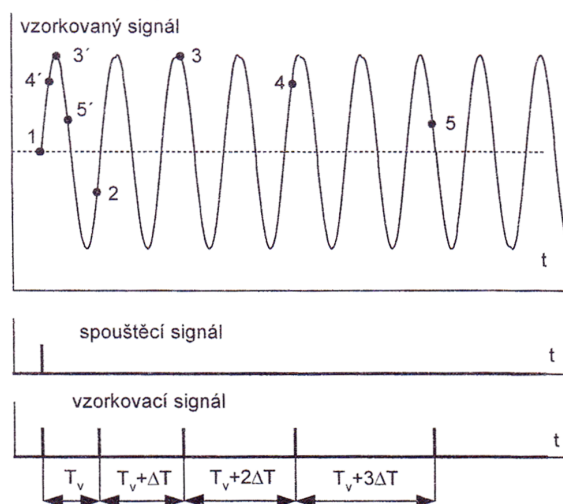


Obr. 1.5: Vzorkování v reálném čase [7]



Obr. 1.6: Sekvenční ekvivalentní vzorkování [7]

3. **Náhodné vzorkování v ekvivalentním čase** (*random repetitive sampling*) je charakterizováno náhodným určením okamžiku odběru vzorku ze vzorkovaného signálu. Vzorkování není synchronizováno se spouštěcím signálem osciloskopu, a proto nemůže dojít k záznějům mezi vstupním a vzorkovaným signálem. Po příchodu spouštěcího impulsu jsou generovány vzorkovací impulsy s náhodným, ale přesně známým časovým rozdělením  $T_1 : T_2$  až  $T_N$ , kde  $N$  je kapacita paměti. Tímto způsobem dostaneme velké množství datových sad, z kterých je rekonstruován původní průběh signálu, obr. 1.7. Předpokladem je opět periodicitu vzorkovaného signálu.[3]



Obr. 1.7: Náhodné ekvivalentní vzorkování [7]

### 1.2.2 Chyby vzorkování

Skutečný vzorkovač pracuje s určitými chybami. Na kvalitu vzorkovače má rozhodující vliv spínač. Po spínači požadujeme minimální odpor v sepnutém stavu, velký odpor v rozepnutém stavu a dostatečnou rychlost spínání (u extrémně rychlých vzorkovačů až v řádu desítek *ps*). V praxi se spínače realizují pomocí unipolárních tranzistorů JFET nebo MOSFET.[3]

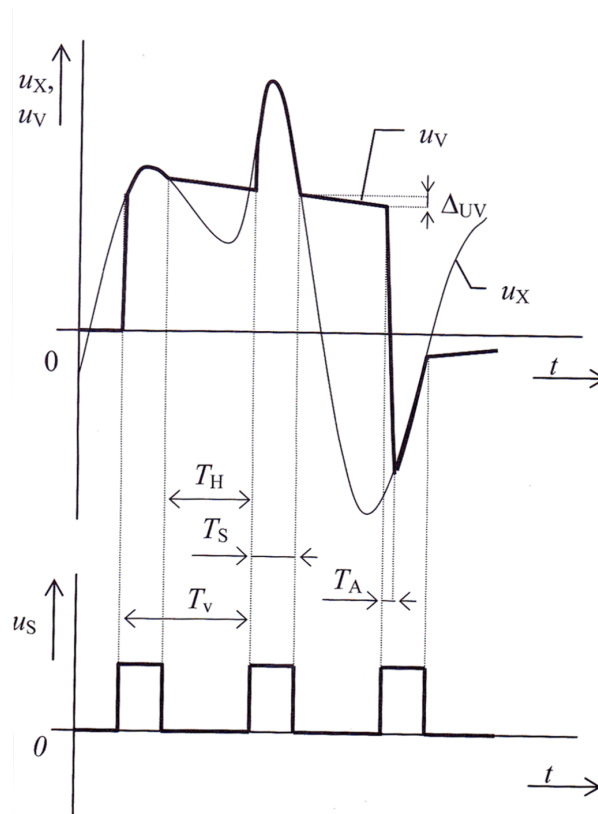
Mezi základní chyby vzorkování patří:

- **Doba upnutí**  $T_A$  (*acquisition time*) je čas, potřebný k přiblížení výstupního napětí k požadované hodnotě s danou chybou, která spadá do předepsaného tolerančního pásma. Výstupní napětí se totiž při přechodu vzorkovače z režimu pamatování do režimu vzorkování nezmění skokem, ale mění se konečnou

rychlostí. Rychlost této změny je dána vlastnostmi zesilovače OZ a časovou konstantou pamatovacího obvodu.[3]

- **Rychlost přeběhu**(*slew rate*) měříme ve  $V/s$ . Udává s jakou maximální rychlostí se dokáže měnit výstupní napětí, pokud na vstup vzorkovače přivedeme napěťový skok.
- **Rychlost změny pamatované hodnoty**(*droop rate*) vyjadřuje změnu výstupního napětí  $\delta_{UV}$  v paměťovém režimu vzorkovače vlivem vybíjení paměťového kondenzátoru.
- **Doba apertury**(*aperture time*) je doba mezi bezprostředním pokynem k rozpojení spínače a ukončením rozpojování, kdy lze spínač považovat za zcela rozpojený. Vzniká v důsledku časových konstant v reálném obvodu.
- **Nejistota apertury**(*aperture uncertainty*) je náhodné kolísání doby apertury. Někdy se označuje jako aperturové chvění (jitter).

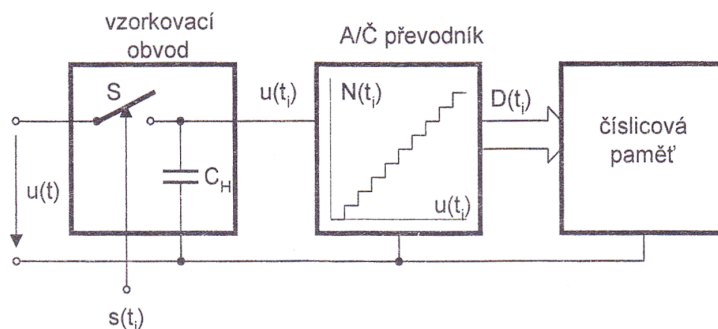
Navíc není výstupní napětí  $u_V$  ani během doby sledování  $T_S$  rovno vstupnímu  $u_X$ . To je dáno offsetem a konečným zesílením reálného operačního zesilovače. Další chyby jako např. doba průniku a chyba přeslechu mají význam pro přesná nebo velmi rychlá měření.[3]



Obr. 1.8: Princip vzorkování v čase [3]

## 1.3 Digitalizace signálu

Řetězec pro digitalizaci spojitého vstupního signálu obsahuje vzorkovací obvod, A/D převodník a paměť dat, obr. 1.9. Vzorkovací obvod byl podrobněji popsán v kapitole 1.2.



Obr. 1.9: Digitalizace spojitého signálu [7]

### 1.3.1 Obecné vlastnosti A/D převodníků

A/D převodník je obvod, který převádí hodnotu vstupního signálu, na jemu odpovídající diskrétní číselnou hodnotu. Při tom dochází ke *kvantování vstupního signálu v úrovních*. Analogově číselový převod spojitého signálu představuje transformaci spojitě funkce  $s(t)$ , popisující vstupní signál na posloupnost čísel  $\{S_n(t_n)\}$ ,  $n = 0, 1, \dots$  vztažených k okamžikům  $t_n = n \cdot T_V$ . Tento proces lze rozdělit do dvou samostatných skupin.[10]

- **Diskretizace** při které dochází k transformaci spojitě funkce na nespojitou

$$s(t) \rightarrow \{S(t_n)\}, \quad (1.6)$$

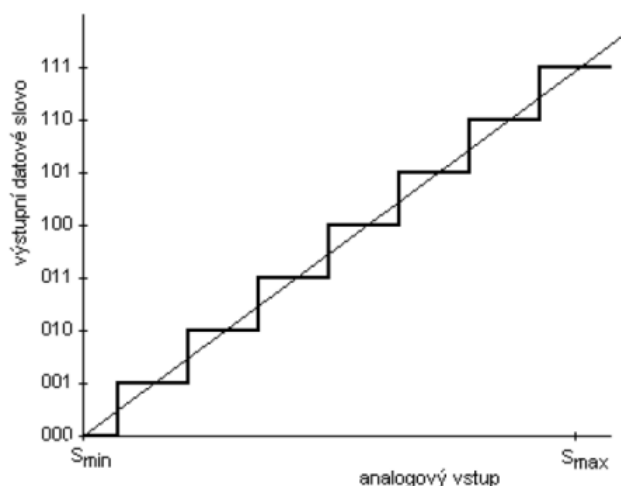
- **Kvantování** což je v podstatě nalezení nejbližšího přípustného číselného ekvivalentu v dostupném rozlišení

$$\{S(t_n)\} \rightarrow \{S_n(t_n)\}. \quad (1.7)$$

Za předpokladů, že se signál  $s(t)$  mění dostatečně pomalu, lze obě operace spojit v jedinou. Ta poté může být realizována přímo A/D převodníkem.

Kvantování znamená vlastně zaokrouhlení vstupní spojitě veličiny  $s$  na nejbližší kvantovací hladinu s odpovídající číselnou vstupní hodnotou  $S$ . A/D převodník nabývá tedy konečné množiny stavů, kterým odpovídá konečný počet kvantovacích

hladin. Vzdálenost kvantovacích hladin je nejčastěji ekvidistantní a nazývá se kvantovací krok  $q$ . Ve výstupním číselném kódu odpovídá  $q$  číselně váze LSB (*least significant bit*), nejméně významného bitu. Převodní charakteristika A/D převodníku je schodovitá funkce. Na 1.10 je uvedena přechodová charakteristika 8bitového A/D převodníku. Ideální charakteristikou je přímka, která prochází středem grafu a odpovídala by převodníku s nekonečně velkým rozlišením.[3]



Obr. 1.10: Převodní charakteristika 8bitového A/D převodníku [10]

Převodníky provádějí kromě kvantování také kódování výstupní číselné hodnoty.

$$S_M \rightarrow N \quad (1.8)$$

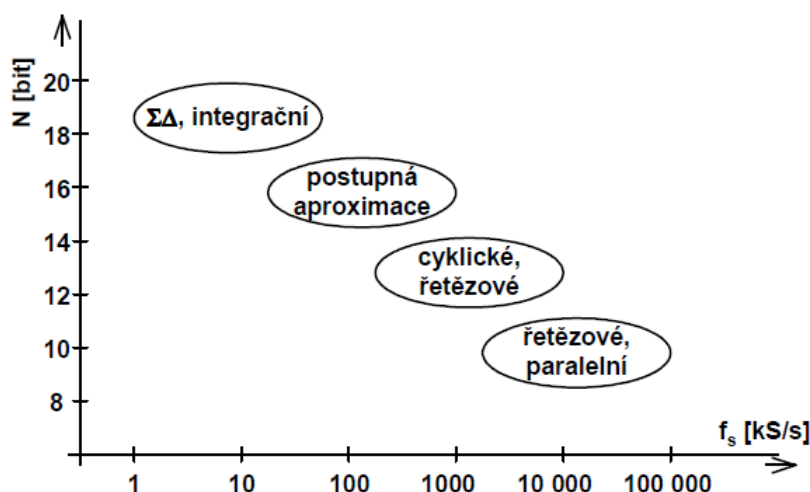
Výstupem převodníku je číselný kód  $N$ . Nejčastěji používané kódy jsou: přímý dvojkový, BCD (dvojkově dekadický), Grayův kód. Méně častěji se u bipolárních A/D převodníků využívá kódů: dvojkový se znaménkem, dvojkový posunutý, inverzní (kód prvního doplňku), dvojkový doplňkový (kód druhého doplňku).[15]

**Rozlišovací schopnost A/D převodníku** je určena počtem úrovní, do kterých je rozdělen rozsah vstupního analogového signálu tzn. počtem kvantizačních úrovní. Výstupní slovo převodníku vyjadřuje obvykle číslo v binárním kódu, protože zpravidla dochází k dalšímu zpracování na zařízeních pracujících v dvojkové soustavě. Kvůli častému užívání binárního kódu je často rozlišovací schopnost vyjadřována počtem bitů výstupního slova. Je-li například výstupní slovo složeno z 10 bitů, pak je výstupní rozsah rozdělen na  $2^{10} = 1024$  diskretních úrovní. Rozlišovací schopnost je  $1/2^N$ , kde  $N$  je počet bitů výstupního slova. S rozlišovací schopností je úzce svázána vzorkovací frekvence. Závislost mezi těmito parametry je dána nepřímou úměrou. U A/D převodníků s vysokou rozlišovací schopností není možné dosáhnout

i vysoké vzorkovací frekvence (četnosti převodu) a naopak. Hodnoty uvedené na obr. 1.11 ukazují parametry nejpoužívanějších A/D převodníků. Existují však i převodníky, které dosahují vysoké rozlišovací schopnosti (až 24 bitů) s vzorkovací frekvencí vysokých řádů (až jednotky GHz).[10]

**Dynamické vlastnosti** se hodnotí podle doby převodu nebo přesněji rychlosti nebo četnosti převodu A/D převodníku. Doba převodu je převrácenou hodnotou rychlosti převodu. Často je doba převodu totožná s dobou vzorkování.[10]

**Četnost nebo rychlost A/D převodů** při zaručené přesnosti převodu určuje maximální počet převodů za jednotku času. Nenulovou dobou vzorkování je způsobená chyba. Vzniká v důsledku změny vstupního analogového signálu během vzorkování. Vyjadřuje se rozdílem hodnot vstupní veličiny na začátku a konci vzorkování. Tuto chybu lze podstatně zmenšit použitím vzorkovače s pamětí popsaného v kapitole 1.2. [10]



Obr. 1.11: Rozdělení ADC podle rozlišení v závislosti na četnosti převodu[10]

Typy A/D převodníků označujeme podle způsobu zjištění číselného ekvivalentu vstupní analogové veličiny. Často používané typy převodníků jsou uvedeny v tabulce 1.1, společně s jejich výhodami a nevýhodami. Pro aplikace, kde je nutná maximální přesnost převodu (zpracování audio signálu) bývají použity převodníky sigma-delta ( $\Sigma - \Delta$ ), nebo převodníky s postupnou aproximací. Pokud je požadován rychlý převod při dobrém rozlišení a nízkou spotřebou (zpracování obrazu) využívají se převodníky řetězové.

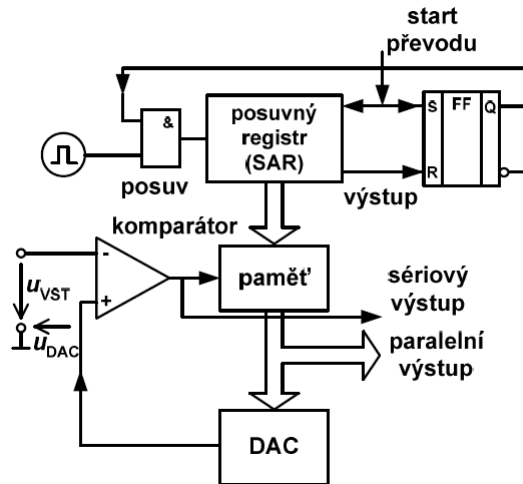
Tab. 1.1: Základní typy převodníků AD - výhody,nevýhody [10]

Typ převodníku	Výhody	Nevýhody
Sigma-delta ( $\Sigma - \Delta$ )	-rozlišovací schopnost až 24 bitů -jednoduché obvodové řešení -nepotřebuje externí S/H obvod díky funkci převzorkování -nepotřebují antialiasingový filtr	-vzorkovací frekvence řádově stovky kHz
integrační	-rozlišovací schopnost až 24 bitů	-vzorkovací frekvence řádově stovky kHz -nutnost velkých kapacit v integrátoru -nákladné na výrobu
postupná aproximace	-rozlišovací schopnost až 20 bitů -malá plocha čipu-nízká spotřeba	-vzorkovací frekvence řádově jednotky MHz -přesnost závisí na přesnosti interního DAC
řetězové	-rozlišovací schopnost až 16 bitů -vysoký vzorkovací frekvence řádově stovky MHz -dosažení validního výsledku převodu ve velmi krátkém čase -plocha čipu roste lineárně s počtem bitů - nízká spotřeba	-přesnost převodu závisí na přesnosti prvních bloků, které řeší váhově nejdůležitější bity -nutnost časového sladení vzorků před odesláním na výstup
paralelní (komparační)	-nejvyšší vzorkovací frekvence až GHz	-nízké rozlišení do 8 bitů -při vyšším rozlišení velká plocha čipu -vysoká spotřeba

### 1.3.2 A/D převodníky s postupnou aproximací

Kompenzační převodníky A/D s postupnou aproximací obsahují řídicí obvod převodníku, který je konstruován pro postupnou aproximaci vstupního napětí. Jako řídicí obvod se v zapojení podle obr. 1.12 využívá posuvný registr SAR (*successive approximation register*). Hodnota měřeného vstupního napětí  $u_{VST}$  se aproximuje vhodně volenými kroky rekonstrukčního napětí  $u_{DAC}$ . Převodník srovnává vstupní napětí postupně s napětími, které odpovídají jednotlivým vahám bitového slova. Postupuje se od bitu s nejvyšší vahou (MSB) až po bit s vahou nejnižší (LSB).[15]

Na začátku cyklu převodu se nastaví hodnota 1 do aproximačního registru na



Obr. 1.12: Kompenzační ADC s postupnou aproximací [10]

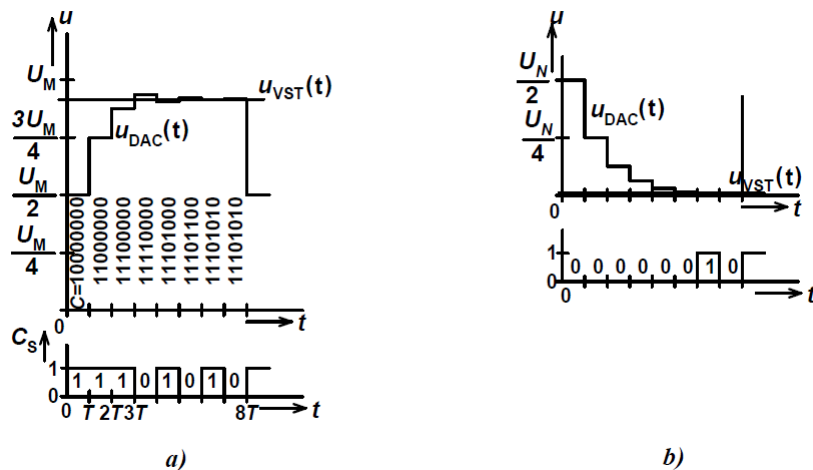
pozici nejvyššího bitu. U 8bitového převodníku tomu na výstupu odpovídá hodnota poloviny maximálního napětí  $u_{M/2}$ . Tato 1 se postupně v dalších krocích posouvá po všech bitech  $n$  – bitového slova  $C$ . Tím se postupně přidávají jednotlivá váhová napětí a komparují se se vstupním napětím převodníku. Podle výstupu komparátoru se na dané pozici bitu jednička buď ponechá ( $u_{VST} > u_{DAC}$ ), nebo se hodnota tohoto bitu změni na nulu ( $u_{VST} < u_{DAC}$ ). Převod v  $n$  bitovém převodníku pro vstupní napětí z rozsahu  $u_{VST} = 0$  až  $u_M$  probíhá vždy právě v  $n$  taktech. Celková doba převodu je konstantní a nezávislá na velikosti vstupního napětí v povoleném rozsahu. Můžeme ji vypočítat podle vzorce 1.9 [10]

$$T_p = n \cdot T \quad (1.9)$$

Pro názornost jsou na obr. 1.13 uvedeny dva grafické průběhy ustalování rekonstrukčního napětí  $u_{DAC}$  pro převod vstupního napětí  $u_{VST}$  blízké 1.13a) maximálnímu rozsahu převodníku. Postupně, počínaje MSB, se vytvořilo ekvivalentní binární číslo 11101010 v  $n = 8$  krocích. Na 1.13b) je naopak vstupní napětí  $u_{VST}$  blízké minimálnímu rozsahu napětí převodníku a výsledkem je dvojkové číslo 00000010.[10]

Hlavní nevýhodou tohoto převodníku je, že změna vstupního napětí během převodu může způsobit hrubou chybu. Proto musí být vstup opatřen vzorkovačem 1.2. Převodníky s postupnou aproximací se běžně vyrábějí jako 8 až 16 bitové s dobou převodu řádově  $\mu s$ .



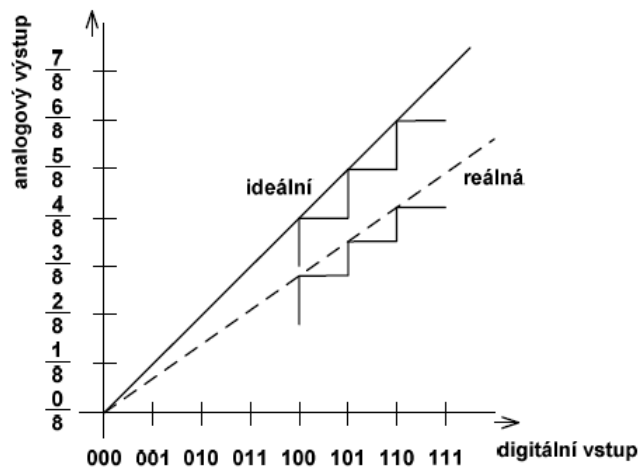


Obr. 1.13: Převod s postupnou aproximací, časový diagram pro a) velké  $u_{VST}$  b) malé  $u_{VST}$  [10]

### 1.3.3 Chyby A/D převodníků

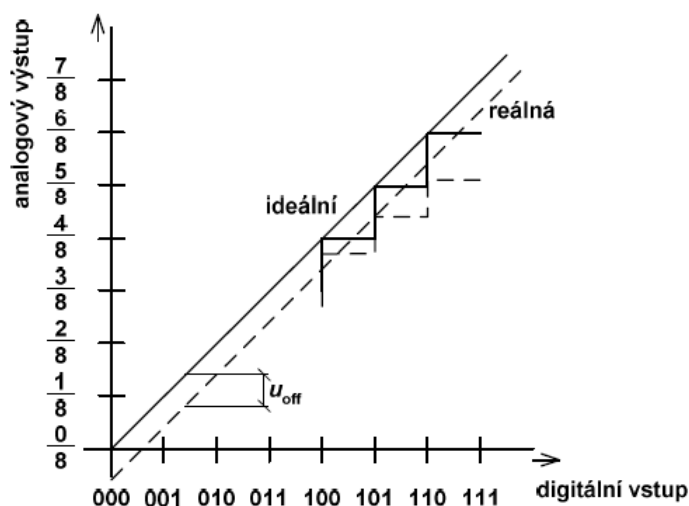
Dosažitelnou přesnost a stálost celého A/D převodníku přímo ovlivňuje zdroj referenčního stejnosměrného napětí  $U_R$ . Ten je obsažen v každém A/D převodníku a bývá realizován elektronickými obvody. Jedná se například o uměle stárnuté, teplotně kompenzované zenerovy diody, v nejpřesnějších přístrojích bývají termostatované. Chyby převodníku samotného lze rozdělit na statické a dynamické. [3]

Základní statické chyby jsou:



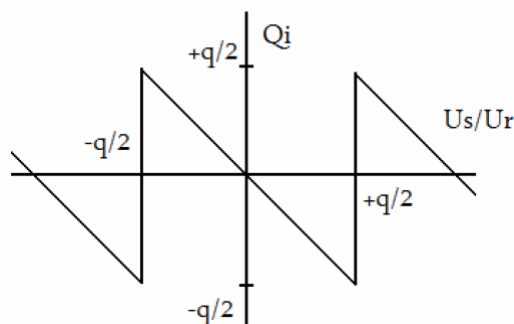
Obr. 1.14: Chyba zesílení A/D převodníku [10]

- **chyba zesílení**, která je dána odchylkou sklonu skutečné převodní charakteristiky A/D převodníku od ideální o určitý úhel, obr. 1.14.
- **chyba nuly**, která je dána posunem počátku převodní charakteristiky ve směru dané osy. Celková charakteristika je poté o daný posuv kratší nebo delší, obr. 1.15.
- **chyba linearity** převodu



Obr. 1.15: Chyba nuly A/D převodníku [10]

**Kvantovací chyba** vzniká důsledkem kvantování. Je zobrazena na obr. 1.16. Jedná se o rozdíl mezi kvantovanou a přesnou hodnotou vzorku. Tyto chyby mají náhodný charakter a v konečném důsledku se projeví jako kvantizační šum. Kvantizační šum nabývá u idealizovaného A/D převodníku velikosti  $\pm q/2$  (polovina kvantizačního kroku). [3]



Obr. 1.16: Kvantovací chyba ideálního A/D převodníku [14]

Efektivní hodnota kvantizačního šumu je dána podle vzorce 1.10, kde  $U_M$  je maximální napětí. Výkon kvantizačního šumu je rovnoměrně rozdělen ve spektru od frekvence 0 do poloviny vzorkovací frekvence.

$$U_{ef} = \frac{q \cdot U_M}{\sqrt{12}} [V] \quad (1.10)$$

Dynamické chyby se uplatní při rychlé změně vstupního napětí. To znamená, že s rostoucím frekvencí vstupního napětí klesá rozlišovací schopnost A/D převodníku. Proto se často u převodníků uvádí tzv. efektivní počet bitů v závislosti na frekvenci.

[3]

## 2 PLATFORMA ARDUINO

Arduino je open-source platforma založena na mikrokontroleru ATmega a grafickém vývojovém prostředí, které vychází z prostředí Wiring (podobná jazyku C) a Processing (prostředí pro urychlení výuky programování původně vyvinuto na MIT). Arduino je schopné snímat okolní prostředí pomocí vstupů z rozličných senzorů. Zároveň může na tyto údaje reagovat připojenými LED diodami, motorky a dalšími výstupními perifériemi. Projekty založené na Arduinu mohou jednoduše komunikovat se softwarem na stolním počítači nebo notebooku (Flash, Processing). Všechny zdrojové soubory jsou k dispozici díky tomu, že je dokumentace a referenční příručka programovacího jazyka a externích knihoven vydávána pod licencí Creative Commons. [4]

### 2.1 Hardwarová část platformy Arduino

Desky Arduino obsahují 8bitové kontrolery z rodiny AVR od firmy Atmel s komplementárními komponenty pro usnadnění programování a začlenění do dalších obvodů. Konektory jsou na deskách standardizovaně rozmístěny a umožňují tak jednoduché připojení přídatných modulů označovaných jako *Shield*. Jedná se např. o Ethernet Shield, Wifi Shield, Motor Shield a jiné. Tyto moduly komunikují s Arduinem přímo pomocí *I/O* pinů nebo jako individuálně adresovatelné přes *I<sup>2</sup>C* sériovou sběrnici, což umožňuje paralelní spojení více *Shieldů*. Na deskách bývá několik diod, resetovací tlačítko, konektory pro ICSP programování, napájecí konektor, 5 V lineární regulátor napětí, 16 MHz krystalový oscilátor a obvod zprostředkovávající komunikaci po USB. Ačkoliv je Arduino připojeno k počítači pomocí rozhraní USB, je softwarově simulována sériová komunikace přes linku RS-232. Hlavní mikrokontrolér, který je uživatelsky programovatelný, již má bootloader (kód, který se po spuštění postará o základní nastavení mikrokontroléru, jako jsou interní časovače, nastavení rozhraní USART a další) a nastavené potřebné fuses bajty (těmi se nízkourovňově nastavují některé vlastnosti čipu). Díky tomu se uživatel nemusí starat o detaily nastavení a své programy píše v jazyce podobném C. [5]

### 2.2 Softwarová část platformy Arduino

Vývojové prostředí (IDE) Arduina je multi-platformová aplikace napsána v programovacím jazyce Java. Je odvozeno z IDE pro Processing programování a prostředí Wiring. Hlavní důraz je kladen na jednoduchost a přehlednost. Editor kódu zahrnuje funkce jako zvýrazňování syntaxe, párování závorek a automatické zarovnávání. Program napsaný pro Arduino se označuje jako *sketch*. [4]

Arduino IDE obsahuje softwarovou knihovnu nazývanou *Wiring*, která originálně pochází ze stejnojmenného projektu. To přináší značné zjednodušení při práci se vstupem i výstupem. Uživatel musí definovat pouze dvě funkce k vytvoření cyklicky běžícího programu.[5] Jedná se o:

- **setup()** je inicializační funkce, která proběhne vždy jednou při spuštění programu.
- **loop()** je funkce, která cyklicky probíhá, dokud nedojde k odpojení napájení desky Arduino. [11]

## 2.3 Arduino Mega ADK

Arduino Mega *Accessory Development Kit*, dále již jen ADK, je vývojová deska založená na ATmega2560. Má Rozhraní USB host pro připojení s telefony Android založené na MAX3421E. Obsahuje 54 digitálních vstupů / výstupů (z toho je 14 použitelných jako výstupy PWM), 16 analogových vstupů, 4 UARTy (sériová komunikace), 16 MHz krystalový oscilátor, rozhraní USB, napájecí konektor, konektor IKSP a resetovací tlačítko. Rozložení výše popsaných součástí je zobrazeno na obr. ?? [4] [12]

### 2.3.1 Vstupní a výstupní napětí

Arduino Mega ADK může být napájeno přes USB nebo pomocí externího zdroje napětí. Napájecí zdroj je vybrán automaticky.

Externí (non-USB) napájení může být realizováno AC-to-DC adaptérem nebo baterií. Adaptér může být k desce připojen pomocí 2,1 mm napájecího konektoru. Baterie může být k desce připojena pomocí pinů Gnd a Vin. Deska je schopná pracovat při externím napájení o velikosti 5,5 V - 16 V. Nicméně pokud je dodáváno méně než 7 V, může být napětí výstupního 5 V pinu podstatně menší. To může způsobit nestabilitu desky. Pokud je napájecí napětí více než 12 V, může docházet k přehřívání desky a jejímu následnému poškození. Proto je doporučený rozsah napájecích napětí 7 V - 12 V.[4]

Napěťové piny jsou následující:

- **VIN** pomocí tohoto pinu je k Arduinu možné připojit externí napájecí zdroj v podobě baterie
- **5V** pin je zdrojem regulovaného napětí hodnoty 5V
- **3V3** pin je zdrojem 3,3 V. toto napětí je vytvářeno integrovaným regulátorem. Maximální odběr je 50 mA
- **GND** zemnicí piny.

- **IOREF** na Arduino desce tento pin poskytuje referenční napětí, s kterým pracuje mikrokontrolér.

### 2.3.2 Paměť

Arduino Mega ADK pracuje s *flash* pamětí o velikosti 256 KB, která slouží pro ukládání kódu. Z toho je vyhrazeno 8 KB paměti pro *bootloader* (zavaděč). Dále deska disponuje 8 KB paměti SRAM a 4KB EEPROM (ke čtení a zápisu slouží knihovna EEPROM).[4]

### 2.3.3 Komunikace

Arduino Mega ADK má řadu zařízení pro komunikaci s počítačem, dalším Arduinem, nebo jiným mikrokontrolérem. ATmega2560 nabízí čtyři hardwarové UART pro TTL (5 V) sériovou komunikaci. Jeden z nich za pomoci ATmega2560 zprostředkovává komunikaci přes USB a poskytuje virtuální COM port pro software počítače (Windows potřebuje navíc .inf soubor, OSX a Linux rozpozná Arduino desku jako COM port automaticky). Software Arduina obsahuje monitor sériového portu (*serial monitor*). Díky němu je možno monitorovat jednoduchá textová data posílaná jak z platformy Arduino, tak na ni. Pokud jsou data přenášena prostřednictvím čipu ATmega2560 mezi platformou Arduino Mega ADK a USB sběrnici počítače, blikají v průběhu procesu LED diody RX a TX.[4]

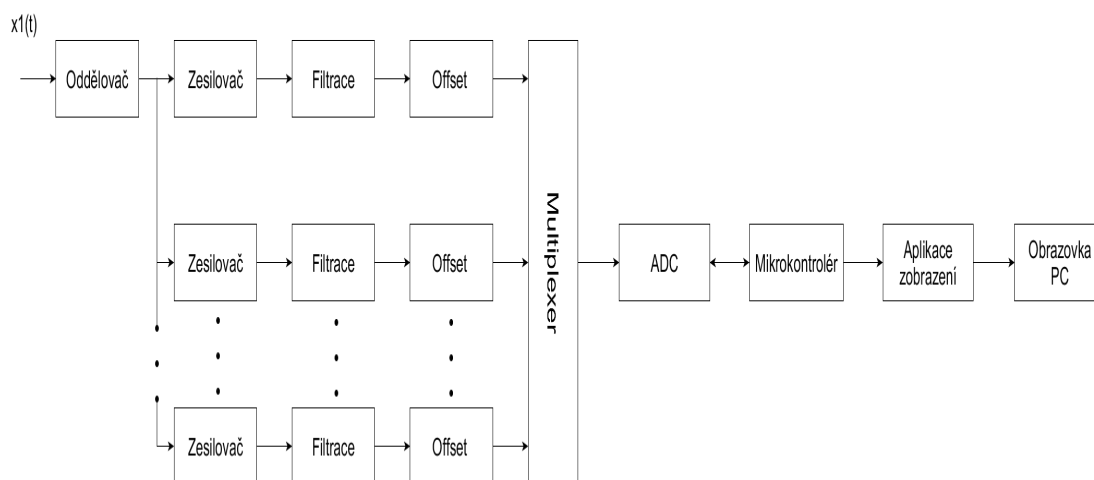
### 2.3.4 Nadproudová ochrana sběrnice USB

Deska Arduino Mega ADK obsahuje resetovatelnou pojistku. Tato pojistka chrání USB porty počítače před zkraty a nadproudy. Přestože v sobě má většina počítačů již zabudovanou vnitřní ochranu, poskytuje tato pojistka přídavnou vrstvu ochrany. Pokud je na sběrnici USB připojen proud větší než 500 mA, dojde k automatickému přerušení spojení, dokud není přetížení či zkrat odstraněn.[4]

## 3 REALIZACE OSCILOSKOPU

### 3.1 Návrh osciloskopu

Prvním krokem při realizaci osciloskopu je vytvoření návrhu, který by splňoval všechny požadavky zadání. Blokového schéma navrhovaného osciloskopu je na obr. 3.1.



Obr. 3.1: Blokové schéma navrhovaného osciloskopu

- **Zesilovač** slouží k zesílení měřeného vstupního napětí. Je tvořen integrovaným obvodem, který obsahuje identické zesilovače, lišící se pouze hodnotou zesílení. Zesílení je odstupňováno  $A_1 = 1$ ,  $A_2 = 10$ ,  $A_3 = 100$ . Obecně může být zesilovačů i více. Aby bylo měřené vstupní napětí minimálně ovlivňováno osciloskopem, musí mít zesilovač vysokou vstupní impedanci. K dalším požadavkům na zesilovač patří široká frekvenční charakteristika (aby co nejméně zkresloval rychlé děje) a nízký vlastní šum. Podrobnější popis je uveden v 4.5.5.
- **Filtr** má za úkol potlačit frekvence vyšší než odpovídají vzorkovacímu teóremu viz. kapitola 1.2. Jedná se o dolnoproustný antialiasingový filtr s mezní frekvencí  $f_{vz}/2$ . Na antialiasingový filtr jsou kladeny vysoké nároky. Při výběru požadujeme co největší strmost a lineární fázovou charakteristiku. Při konstrukci filtry nebyly použity, aby nedošlo k změně vstupního signálu. Uživatel musí pro správnou funkci osciloskopu měřit signál o frekvenci splňující vzorkovací teorem.
- **Offset** je přičten ke vstupnímu signálu. Zajišťuje, že rozsah vstupních napětí do platformy Arduino Mega ADK bude v přípustných mezích. Realizace je rozepsána v kapitole 3.3.2

- **Multiplexer** je použit pro přepínání mezi zesilovači s různým zesílením. Zesílené signály jsou přivedeny na analogové piny platformy Arduino Mega ADK a pro přepínání mezi nimi je použit integrovaný multiplexer.
- **ADC** se stará o převod spojitého (analogového) signálu na signál diskrétní (digitální). Je součástí desky Arduino ADK. Vlastnosti ADC (rozlišovací schopnost, maximální vzorkovací frekvence) jsou podrobněji rozebrány v kapitole 4.3
- **Mikrokontrolér** je součástí desky Arduino ADK. Jedná se o ATmega2560. Způsob jeho naprogramování včetně vývojového diagramu je uveden v kapitole 3.2.1.
- **Aplikace zobrazení** přijímá data odeslaná z mikrokontroléru a zpracovává je tak, aby je bylo možné vykreslit na obrazovku PC. Je díky ní možné měnit parametry akvizice dat. Podrobnější popis aplikace včetně vývojového diagramu je uveden v kapitole 3.2.2
- **Obrazovka PC** slouží pro zobrazení naměřených průběhů.

## 3.2 Software osciloskopu

Softwarová část vytvořeného osciloskopu se skládá ze dvou vzájemně komunikujících programů. První z nich, program mikrokontroléru, obstarává akvizici dat. Druhý, aplikace zobrazení, naměřená data přijímá, zpracovává, a vhodným způsobem zobrazuje na obrazovce počítače, dále již jen PC. K tomuto účelu bylo vytvořeno grafické uživatelské rozhraní, dále již jen GUI.

Původním záměrem bylo spojit platformu Arduino Mega ADK se zařízením, které podporuje operačním systémem Android. I přes dlouhotrvající snahu se vzájemnou komunikaci nepodařilo navázat. Byl vyzkoušen návod na oficiálních stránkách Arduino Lab[4], jehož poslední aktualizace proběhla v roce 2010. Dále byl vyzkoušen nástroj Tellart, aplikace Handbag, knihovny vytvořené samotnými uživateli dostupné na GitHub (dokumentace je nedostatečná nebo zcela chybí) a další způsoby. Taktéž snaha kontaktovat výrobce byla bez úspěchu. Problémy při spojení mohou být způsobeny změnou komunikačního protokolu u operačního systému Android verze 4.1. Po konzultacích a dohodě s vedoucím práce bylo GUI vytvořeno v prostředí Matlab.

Nejdříve byl pro akvizici a přenos dat mezi Arduinem a Matlabem použit softwarový balíček vyvinut konkrétně pro jejich vzájemnou komunikaci firmou MathWorks. Na straně platformy Arduino Mega ADK bylo pro funkci příkazů nutné nahrát předpřipravený program. V prostředí Matlab pak doinstalovat bezplatně poskytovanou knihovnu. Po provedení těchto kroků umožňoval Matlab čtení analogového portu platformy Arduino příkazem *analogRead*. Velkou nevýhodou tohoto způsobu akvi-



zice dat byla jeho rychlost. Nejvyšší vzorkovací frekvence se pohybovala okolo 20 Hz.

Rychlost akvizice dodávaného software byla tedy pro účely osciloskopu nedostatečná. Proto se další práce zabývala vyvinutím vlastního programového vybavení pro akviziční část (program mikrokontroléru) a aplikaci zobrazení (program pro PC).

### 3.2.1 Program akviziční části

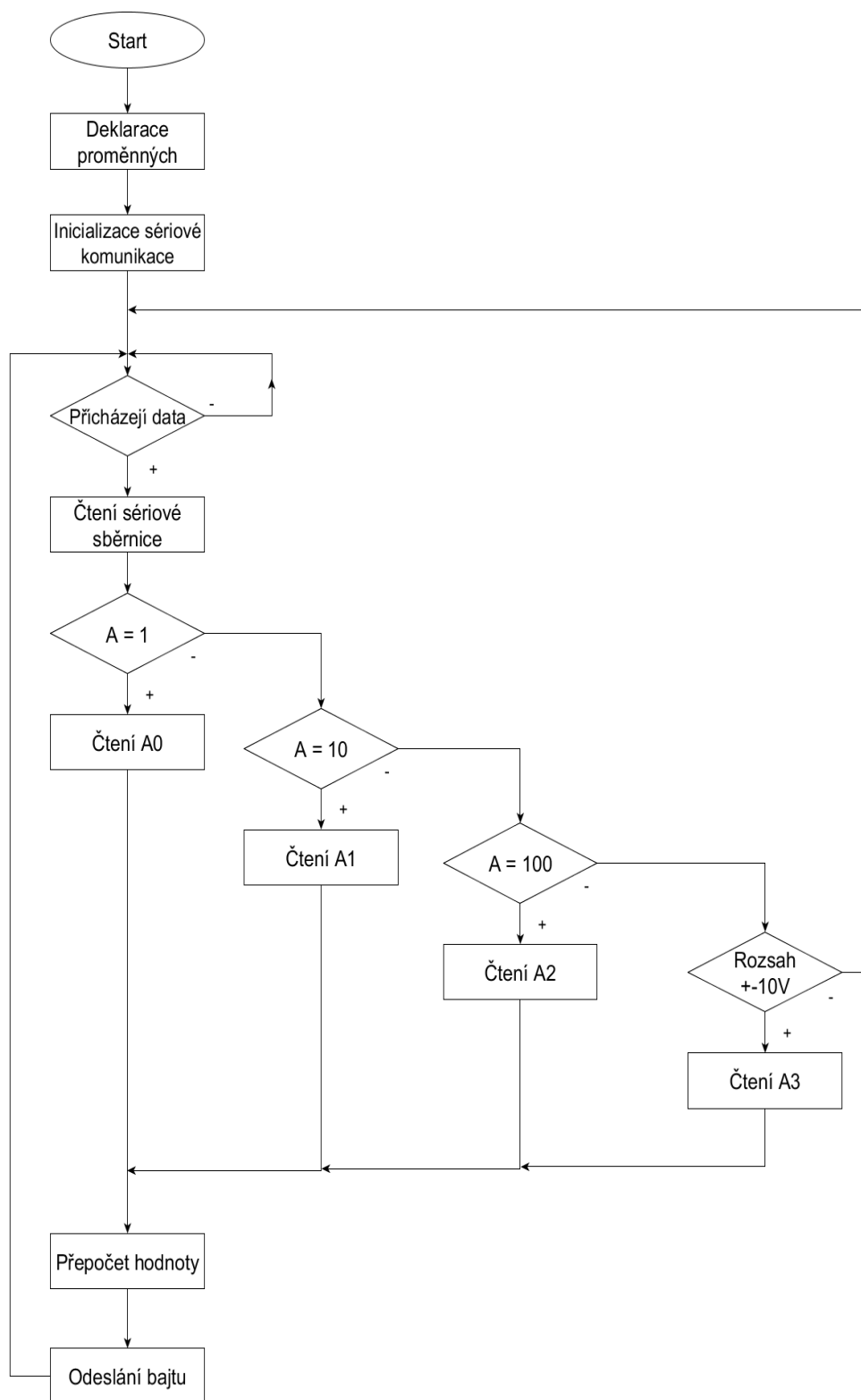
Tvorba programu akviziční části probíhala v integrovaném vývojovém prostředí platformy Arduino. Hlavním požadavkem bylo co možná nejrychlejší čtení analogových pinů a tím také dosažení maximální rychlosti vzorkování měřeného vstupního signálu. Taktéž bylo nutno zajistit reakci programu na požadavky uživatele přicházející z GUI na straně PC. Důležitým faktorem pro bezproblémovou komunikaci obou programů se ukázal být formát odesílaných dat.[11]

- **První verze**

První verze programu akviziční části nepřetržitě posílala data v nekonečné smyčce vzorek po vzorku. Tímto způsobem se dosáhlo maximální vzorkovací frekvence vývojového kitu Arduina Mega ADK. Datový tok vzorků odesílaný po sériové lince v datovém typu integer (16bitový) ovšem nedokázal Matlab funkcí *fscanf* dostatečně rychle zpracovávat. Díky tomu byl zanedlouho naplněn vstupní buffer a docházelo ke ztrátě nejen vstupních dat, ale i časové informace odesílaných vzorků. Čas vzorků byl měřen od doby odeslání do přečtení aplikací zobrazení v PC. Vzhledem k tomu, že nebylo možno určit, jak dlouho data ve vstupním bufferu setrvávají, než jsou přečtena, neodpovídal zobrazovaný signál časově měřenému průběhu.

- **Druhá verze**

Ve druhé verzi probíhalo vzorkování vstupních dat stejným způsobem jako v první. Společně s hodnotou vzorku byl ale odesílán také čas vzorku měřený na straně Arduina funkcí *micros*. Datový tok byl nyní složen ze vzorků v datovém typu integer a časovém údaji v datovém typu long (32bitový). Data byla ze sériové sběrnice opět čtena funkcí *fscanf*. Taktéž docházelo ke ztrátě dat při naplnění vstupního bufferu a díky přidání informace o času vzorku rychleji. Zobrazovaný průběh nicméně nyní časově korespondoval se vstupním měřeným signálem. Tento stav byl dlouho považován za konečný. Při jakémkoliv jiném oddělení vstupních dat do PC (mezerou, velkým písmenem atp.) než znakem nového řádku byla data sice čtena rychleji, nicméně docházelo k jejich deformacím (hodnoty vzorku a času byly nerozeznatelné, často spojeny).



Obr. 3.2: Vývojový diagram akviziční části

- **Třetí verze**

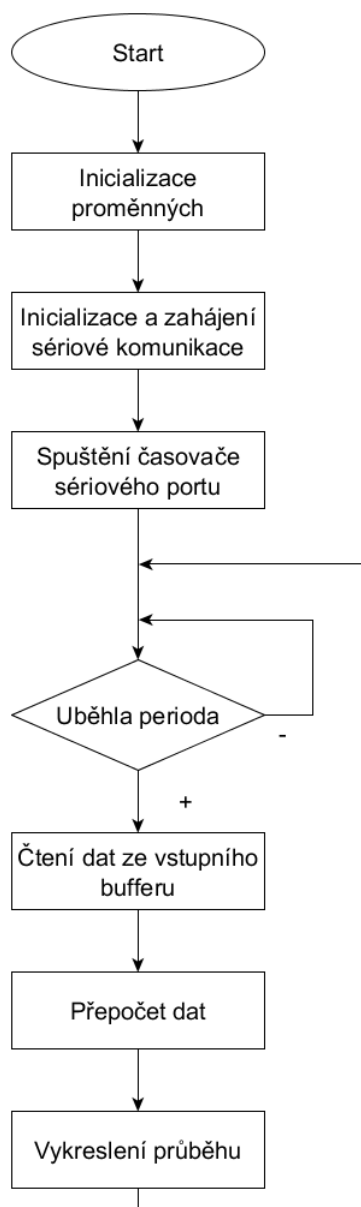
Jako řešení těchto problémů se nakonec ukázalo přistupovat k datům jako k jednotlivým bajtům a periodické čtení sériového portu pomocí časovače sériového portu. Vývojový diagram konečné verze programu akviziční části je na obr. 3.2.

Nejprve jsou proměnným přiřazeny příslušné datové typy podle jejich funkce. Sériová komunikace je zahájena rychlostí 115200 baud. Je důležité, aby rychlost sériové komunikace byla stejná jak na straně platformy Arduino Mega ADK, tak v prostředí Matlab na straně PC. Poté jsou kontrolovány přicházející data podle stisknutých tlačítek v GUI. Při stisknutí tlačítka Zesílení 1,10 nebo 100 je čten příslušně zesílený signál z analogového pinu A0, A1 nebo A2. V případě, že uživatel tlačítkem zvolí rozsah  $\pm 10$  V, je čten analogový vstup A3 s výchozím zesílením 1, které v daném rozsahu nelze změnit. Hodnota vzorku vstupního analogového signálu je z integrovaného A/D převodníku odesílána v 10bitovém rozlišení. Maximální hodnota A/D převodníku je  $2^{10} - 1 = 1023$ . Modul sériové komunikace posílá do PC bajty dat  $2^8 - 1 = 255$ . Přepočítání odesílané hodnoty je tedy  $(255/1023) * v$ , kde  $v$  je hodnota vzorku. Takto upravená hodnota je odeslána příkazem *Serial.write*. Díky tomuto způsobu vzroste množství přenesených dat. V aplikaci zobrazení dojde po obdržení hodnoty k inverznímu přepočtu.

### 3.2.2 Aplikace zobrazení

Jedná se o GUI vytvořené v interaktivním vývojovém prostředí a skriptovacím jazyce čtvrté generace Matlab. Aplikace zobrazení přijímá data, která jsou posílána z programu akviziční části prostřednictvím sériového portu. Přijímaná data dále zpracovává a vykresluje průběh napětí v závislosti na čase. Vývojový diagram aplikace zobrazení je na obr. 3.3

Na začátku programu jsou proměnné inicializovány na výchozí hodnoty. Velikost vstupního bufferu byla při testování programu stanovena na 1024 bajtů. Při této velikosti nedochází ke ztrátě dat a nároky na paměť jsou minimální. Perioda časovače sériového portu je nastavena na nejnižší možnou hodnotu 0.001 s. Při spuštění GUI je zobrazen graf s výchozí délkou časové osy 10 s, napěťovém měřítku 10 V/dílek a časovém měřítku 1 s/div. Po zadání čísla sériového portu následuje specifikace datového toku: počet datových bitů 8, stop bit 1, bez parity, přenosová rychlost sériového rozhraní 115200 baud, která odpovídá přenosové rychlosti programu akvizice dat. Čtení dat je prováděno díky časovači sériového portu v periodických intervalech voláním funkce *scandataGUI*. V každé periodě je přečteno 500 bajtů, které jsou následně graficky zobrazeny funkcí *stripchart*. Čtení a zobrazení dat tedy neprobíhá



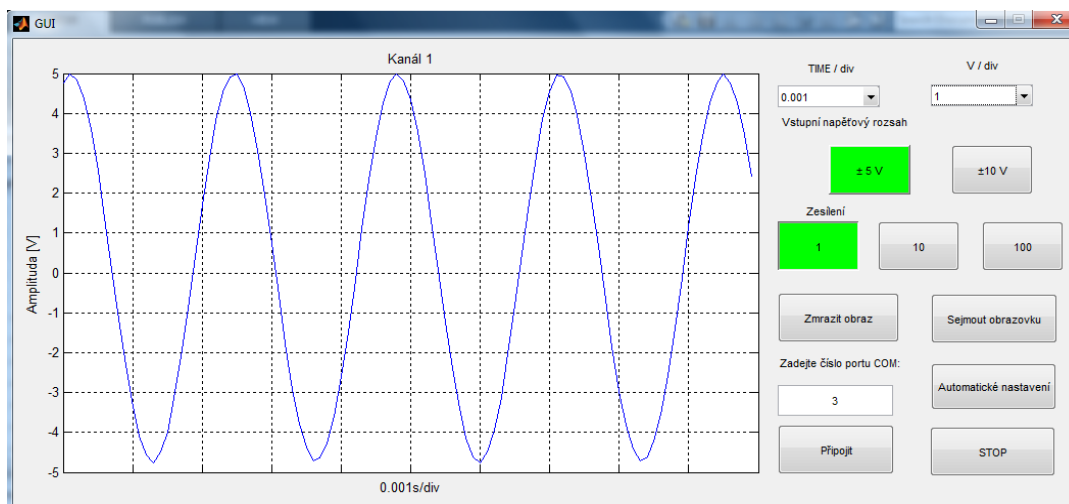
Obr. 3.3: Vývojový diagram aplikace zobrazení

v reálném čase, ale s periodou 1 ms .

Funkce *stripchart* je součástí volně dostupného rozšíření prostředí Matlab s názvem *Test and Measurement Widgets*, jehož autorem je Michelle Hirsch. Slouží pro zobrazení dat, která přichází po sériovém portu. Aby tato funkce lépe vyhovovala aplikaci osciloskopu, byl její zdrojový kód upraven. Nevyhovující byla samovolná dynamická změna napěťové osy, spolu s jejím označením.

Na obr. 3.4 je ukázka GUI, jehož prostřednictvím může uživatel ovlivňovat průběh zpracování měřeného analogového signálu.

- **COM port** očekává vyplnění editovatelného okna číselnou hodnotou portu



Obr. 3.4: Grafické uživatelské rozhraní

COM, ke kterému je připojeno Arduino Mega ADK. Výběr je nutno potvrdit stiskem klávesy ENTER. Tuto hodnotu je možno pro operační systém Windows nalézt v nabídce *Správce zařízení*.

- **Připojit** je tlačítko sloužící v případě validně vyplněného portu COM k vymazání aktivně otevřených portů a časovačů. Následně je připojena platforma Arduino Mega ADK a spuštěn časovač seriového portu společně s jeho čtením.
- **TIME/div** nabízí formou rozbalovacího menu možnost změny časového měřítka na hodnoty: 2 s/div, 1 s/div, 0,5 s/div, 0,1 s/div, 0,05 s/div, 0,01 s/div, 0,005 s/div, 0,001 s/div, 0,0005 s/div.
- **V/div** taktéž umožňuje pomocí rozbalovacího menu měnit napěťové měřítko na hodnoty: 5 V/div, 2 V/div, 1 V/div, 0,5 V/div, 0,1 V/div, 0,05 V/div.
- **Vstupní napěťový rozsah** je možno zvolit stisknutím tlačítka  $\pm 10$  V. V tomto případě je automaticky nastaveno zesílení jedna, jelikož jiné hodnoty zesílení tento rozsah neumožňuje. Tlačítko  $\pm 5$  V má pouze informační funkci, jelikož v tomto rozsahu probíhají všechna měření s jedinou výjimkou stisknutí tlačítka  $\pm 10$  V.
- **Zesílení** je možné ovládat třemi tlačítky a totiž:  $A = 1$ ,  $A = 10$ ,  $A = 100$ . Analogový signál vstupující do platformy Arduino Mega ADK, je pak touto hodnotou zesílen. Tím je umožněno měření i malých signálů v řádu desítek milivoltů. Vybrané zesílení je indikováno změnou barvy tlačítka.
- **Zmrazit obraz / Pokračovat** způsobí při prvním stisku tlačítka dočasné přerušení vykreslování čtených dat do grafu. Zobrazen je statický průběh dat vykreslených před přerušením zobrazení. Data jsou i nadále čtena a ukládána do vstupního bufferu, po jeho naplnění ovšem dochází k jejich ztrátě. Druhý

stisk tlačítka má za následek vyprázdnění bufferu a následnou akvizici dat v běžném režimu.

- **Sejmout obrazovku** uloží aktuální obrazovku GUI do adresáře, ze kterého bylo spuštěno. Uložený snímek je ve formátu .emf<sup>1</sup>.
- **Automatické nastavení** napětového měřítka je provedeno vyhodnocením velikosti vstupních dat. Podle toho, do jakého rozsahu napětí aktuálně měřená data spadají, je nastaven příslušný hodnota V/div. Problematika automatického nastavení časového měřítka je komplikovaná, o čemž svědčí i fakt, že laboratorní osciloskopy mnohdy nejsou schopny tuto hodnotu správně nastavit. Při měření periodických průběhů by bylo možno několik period signálu uložit a podle rychlosti opakování změnit časové měřítko. Obecně ale periodicitu průběhů nelze očekávat.
- **STOP** tlačítko náležitě ukončí aktivně otevřený sériový port, časovač sériového portu a taktéž oba objekty smaže. Je také zavřeno GUI.

### 3.2.3 Komunikace

Obecně lze ke komunikaci mezi Arduinem Mega ADK a jiným zařízením využít univerzální asynchronní přijímač/vysílač, dále již jen UART. Knihovna *Serial* je standardní součástí vývojového prostředí platformy Arduino. Používá hardwarový UART, který je na pinech RX a TX. Tyto piny jsou zároveň připojeny k čipu ATmega2560 a tím pádem slouží i jako sériová linka při spojení s PC. Při používání příkazů z knihovny je automaticky použit sériový komunikační protokol, což napomáhá zjednodušit obousměrnou komunikaci platformy Arduino s PC. Maximální přenosová rychlost je 115200 baud, přičemž je možné přenést při 10bitovém vzorku rozlišení 350 sps.

Pokud bychom netrvali na 10bitovém rozlišení A/D převodníku, je možno zvýšit vzorkovací frekvenci změnou hodnoty děličky. V tomto případě již ovšem sériová komunikace nedisponuje dostatečnou rychlostí a docházelo by ke ztrátě dat. Jiné možné zajištění vzájemné komunikace platformy Arduino s PC je uvedeno v následujícím výčtu.

Další možnosti komunikace Arduina s PC:

- **I<sup>2</sup>C**

Sběrnice I<sup>2</sup>C rozděluje zařízení na řídicí (master – zahajuje a ukončuje komunikaci; generuje hodinový signál SCL) a řízené (slave – zařízení adresované masterem). U Arduina Mega ADK slouží pro I<sup>2</sup>C komunikaci piny SDA (tok dat), SDC (hodinový signál). Připojení s PC je možné například pomocí FTDI

---

<sup>1</sup>Formát grafických souborů enhanced metafile

čipu FT201X, který zprostředkovává plno rychlostní komunikaci USB do I<sup>2</sup>C. Tento čip byl bohužel jen krátce zapůjčen. Rychlost přenosu dat je v tomto případě až 3,4 mbps a plně by pokryla přenos dat vzorkovaných při nastavení nejnižší hodnoty prescaler 16 a rozlišení 8bitů.

- **Bluetooth**

Jedná se o bezdrátovou komunikaci spojující dvě a více zařízení. Její výhodou je větší rychlost přenosu dat a jejich bezdrátový přenos. Nevýhodou u platformy Arduino Mega ADK je nutnost použití externího bluetooth modulu, spotřeba energie a v porovnání se simulovanou sériovou komunikací složitá implementace komunikačního protokolu.

- **WiFi, Ethernet**

Komunikace prostřednictvím WiFi nebo Ethernet přesahuje požadavky aplikace. Oba způsoby vyžadují zakoupení příslušných rozšiřujících modulů.

### 3.3 Hardware osciloskopu

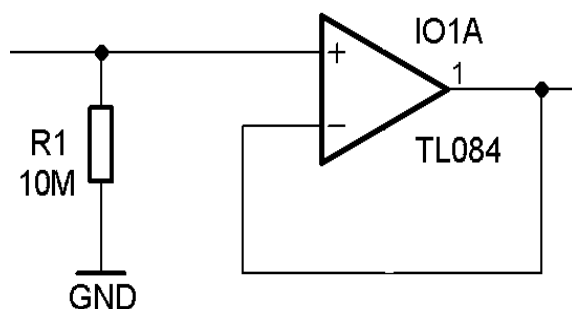
Obvod předřazený platformě Arduino Mega ADK zajišťuje vhodnou úpravu měřeného vstupního analogového signálu před vzorkováním. Mezi požadavky patřilo: minimální ovlivnění měřeného obvodu připojením osciloskopu, měnitelné zesílení vstupního signálu, rozsah měřeného napětí  $\pm 5$  V, snadné napájení.

Při tvorbě hardware byl využit simulační program elektrických obvodů NI Multisim 13.0. Po testování obvodu prostřednictvím simulace, byl obvod zapojen na nepájivé pole, kromě napájení, které bylo nahrazeno laboratorními zdroji. Jeho funkce byla ověřena generátorem funkcí. Kompletní schéma zapojení bylo vytvořeno v programu Eagle a je uvedeno v příloze A.

#### 3.3.1 Zesílení vstupního signálu

Pro zesílení vstupního signálu byl vybrán operační zesilovač, dále již jen OZ, TL084 na základě: vysoké vstupní impedance, velké rychlosti přeběhu 13 V/ $\mu$ s a malé spotřeby 1,4 mA/kanál. Integrovaný obvod obsahuje čtveřici stejných operačních zesilovačů a je napájen  $\pm 12$  V.[16] Podrobněji bude o napájení pojednáno v kapitole 3.3.4

První OZ TL084, v kompletním schématu zapojení označen jako IO1A (analogicky jsou označeny všechny integrované obvody), je zapojen jako napěťový sledovač se zesílením  $A_1 = 1$  a předřazeným odporem  $R_1 = 10$  M $\Omega$ . Schéma zapojení OZ je na obr. 3.5. Jedná se o výřez z celkového schéma zapojení, které je uvedeno v příloze A. Vysokou vstupní impedancí je zajištěno minimální ovlivňování měřeného obvodu po připojení osciloskopu.

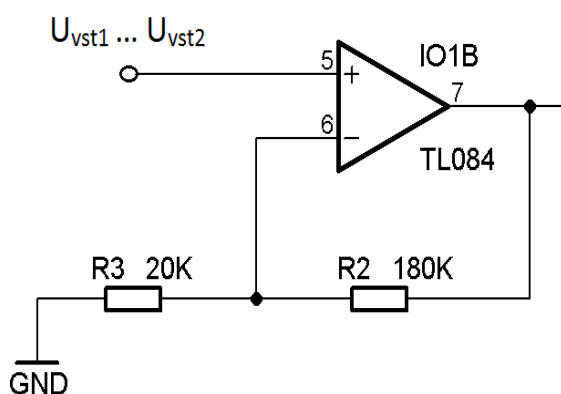


Obr. 3.5: Zapojení vstupního členu - napěťový sledovač

Druhý a třetí OZ TL084, označené IO1B a IO1C, jsou zapojeny jako neinvertující operační zesilovače se zesílením  $A_2 = 10$ ,  $A_3 = 100$  jsou na obr. 3.6. Zapojení se liší pouze hodnotou odporů zpětné vazby[6]. Odporů byly vypočítány podle vztahu:

$$U_{výst} = U_{vst} \cdot \left(1 + \frac{R_3}{R_2}\right) \quad (3.1)$$

Vypočítané hodnoty odporů jsou:  $R_2 = 180 \text{ k}\Omega$ ,  $R_3 = 20 \text{ k}\Omega$ ,  $R_4 = 200 \text{ k}\Omega$ ,  $R_5 = 2 \text{ k}\Omega$ . Poslední OZ TL084, označen jako IO1D, není zapojen.

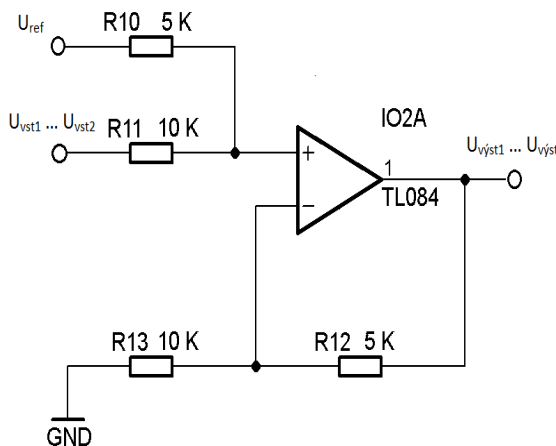


Obr. 3.6: Zapojení neinvertujícího zesilovače



### 3.3.2 Offset vstupního signálu

Analogové piny platformy Arduino Mega ADK, dokáží pracovat se vstupním signálem o velikosti 0 V - 5 V a napětovém rozlišení mezi údaji  $5/1024 = 4,9$  mV při zesílení  $A_1 = 1$ . Pro měření menších signálů slouží zesílení  $A_2 = 10$  a  $A_3 = 100$ . K tomu, aby bylo možné měřit napětí kladná i záporná v rozsahu  $\pm 5$  V, je využit druhý integrovaný obvod. Jedná se opět o čtyři OZ TL084 v jednom pouzdru[16]. Všechny jsou zapojeny jako neinvertující sumační zesilovače více obr. 3.7.



Obr. 3.7: Zapojení sumačního OZ

Přicházející signál je zesílen hodnotou  $A = 0,5$  a je k němu přičtena pevná napětová reference 2,5 V. Do IO2B, IO2C, IO2D vstupují předchozím stupněm IO1A - IO1C zesílené signály v napětovém rozsahu  $\pm 5$  V. Výstupní napětový rozsah je 0 - 5 V přičemž:

$$U_{vst} = -5 \text{ V odpovídá } U_{vyst} = 0 \text{ V}$$

$$U_{vst} = 0 \text{ V odpovídá } U_{vyst} = 2,5 \text{ V}$$

$$U_{vst} = +5 \text{ V odpovídá } U_{vyst} = +5 \text{ V}$$

Při výpočtu odporů vycházíme z rovnice pro výstupní napětí sumačního zesilovače:

$$U_{vyst} = \left( U_{vst} \cdot \frac{R_{10}}{R_{11} + R_{12}} + U_{ref} \cdot \frac{R_{11}}{R_{11} + R_{12}} \right) \cdot \left( 1 + \frac{R_{12}}{R_{13}} \right) \quad (3.2)$$

Zapíšeme-li tuto rovnici pro dva případy vstupního napětí  $U_{vst1}$ , což je dolní hranice vstupního rozsahu a  $U_{vst2}$ , která značí horní hranici vstupního rozsahu, dostaneme:

$$U_{vyst1} = \left( U_{vst1} \cdot \frac{R_{10}}{R_{11} + R_{12}} + U_{ref} \cdot \frac{R_{11}}{R_{11} + R_{12}} \right) \cdot \left( 1 + \frac{R_{12}}{R_{13}} \right) \quad (3.3)$$

$$U_{výst2} = \left( U_{vst2} \cdot \frac{R_{10}}{R_{11} + R_{12}} + U_{ref} \cdot \frac{R_{11}}{R_{11} + R_{12}} \right) \cdot \left( 1 + \frac{R_{12}}{R_{13}} \right) \quad (3.4)$$

Dostaneme dvě rovnice o čtyřech neznámých. Ve skutečnosti nepotřebujeme vypočítat hodnotu každého rezistoru, ale pouze jejich poměr. Použijeme-li substituci  $k_1 = \frac{R_{10}}{R_{11}}$  a  $k_2 = \frac{R_{12}}{R_{13}}$ . Dostaneme:

$$U_{výst1} = \left( U_{vst1} \cdot \frac{k_1}{1 + k_1} + U_{ref} \cdot \frac{1}{1 + k_1} \right) \cdot (1 + k_2) \quad (3.5)$$

$$U_{výst2} = \left( U_{vst2} \cdot \frac{k_1}{1 + k_1} + U_{ref} \cdot \frac{1}{1 + k_1} \right) \cdot (1 + k_2) \quad (3.6)$$

Vydělíme-li první rovnici druhou, eliminujeme z rovnic  $k_2$  a po úpravě bude  $k_1$  rovno:

$$k_1 = U_{ref} \cdot \frac{U_{výst1} - U_{výst2}}{U_{výst2} \cdot U_{vst1} - U_{výst1} \cdot U_{vst2}} \quad (3.7)$$

Po dosazení tohoto výrazu do první rovnice bude  $k_2$ :

$$k_2 = \frac{U_{výst1} - U_{výst2}}{U_{vst1} - U_{vst2}} + \frac{U_{výst2} \cdot U_{vst1} - U_{výst1} \cdot U_{vst2}}{U_{ref} \cdot (U_{vst1} - U_{vst2})} - 1 \quad (3.8)$$

Nyní známe poměry rezistorů  $k_1$  a  $k_2$ . Pro výpočet hodnoty rezistorů  $R_{10}, R_{12}$  zvolíme  $R_{11} = 10\text{k}\Omega$  a  $R_{13} = 10\text{k}\Omega$

Finální rovnice po zpětné substituci za  $k_1$  a  $k_2$ :

$$R_{10} = R_{11} \cdot U_{ref} \cdot \frac{U_{výst1} - U_{výst2}}{U_{výst2} \cdot U_{vst1} - U_{výst1} \cdot U_{vst2}} \quad (3.9)$$

$$R_{12} = R_{13} \cdot \frac{U_{výst1} - U_{výst2}}{U_{vst1} - U_{vst2}} + \frac{U_{výst2} \cdot U_{vst1} - U_{výst1} \cdot U_{vst2}}{U_{ref} \cdot (U_{vst1} - U_{vst2})} - 1 \quad (3.10)$$

Pro požadovanou aplikaci je  $U_{vst1} = -5\text{ V}$ ,  $U_{vst2} = +5\text{ V}$ ,  $U_{výst1} = 0\text{ V}$  a  $U_{výst2} = 5\text{ V}$ . Pro zvolené referenční napětí  $U_{ref} = 2,5\text{ V}$  a rezistory  $R_{11} = 10\text{ k}\Omega$  a  $R_{13} = 10\text{ k}\Omega$  dosazením do rovnic 3.9 a 3.10 vypočítáme:

$$R_{10} = 5\text{k}\Omega \text{ a } R_{12} = 5\text{k}\Omega$$

Výpočet byl proveden pomocí výpočetního software Maple 17. Rozlišení mezi napěťovými údaji klesne o polovinu  $10/1024 = 9,8\text{ mV}$ . [13]

Jako rozšiřující vstupní napěťový rozsah je zapojen čtvrtý OZ TL084, s označením IC2A. Jeho vstupem je výstupní signál IC1A (zesílení  $A_1 = 1$ ). Hodnoty odporů jsou vypočítány podle výše uvedených vzorců pro rozsah vstupního napětí  $U_{vst} = \pm 10\text{ V}$ . Výstupní napětí bude jako u ostatních sumačních zesilovačů  $U_{výst} = 0 - 5\text{ V}$ . Díky tomu je teoreticky vstupní napěťový rozsah zdvojnásoben. Rozlišení mezi

napětovými údaji klesne na  $20/1024 = 19,5$  mV při zesílení  $A_1 = 1$ . Jiné hodnoty zesílení pro rozsah  $\pm 10$  V, kvůli vnitřní konstrukci nelze zvolit. Odpory byly vypočítány způsobem popsaným výše. Při stejném značení odporů jako na obr. 3.7 vyšly pro vstupní napětový rozsah  $\pm 10$  V hodnoty:  $U_{ref} = 2,5$  V,  $R_{11} = 10$  k $\Omega$ ,  $R_{13} = 10$  k $\Omega$ ,  $R_{10} = 2,5$  k $\Omega$  a  $R_{12} = 2,5$  k $\Omega$

### 3.3.3 Koncový stupeň

Koncový stupeň slouží jako ochrana platformy Arduino ADK před poškozením. K tomu dojde při přivedení hodnoty větší než 5,5 V na jakýkoliv vstupní analogový pin. Jelikož jsou všechny IO kromě koncového stupně napájeny napětím  $\pm 12$  V, více v kapitole 3.3.4, došlo by k poškození platformy Arduino Mega ADK v případech přivedení napětí: přesahující vstupní rozsah, způsobující nasycení IO1B a IO1C.[16]

Je-li například na vstup osciloskopu přivedeno napětí o amplitudě  $U_{PK-PK} = 0,3$  V, došlo by po jeho zesílení v IO1C ( $A_3 = 100$ ) k dosažení stavu saturace OZ  $U_{výst_{IO1C}} = \pm 12$  V. Na vstupu sumačního OZ by poté vstupní napětí neodpovídalo očekávanému vstupnímu rozsahu. Důsledkem by bylo výstupní napětí přesahující rozsah 0 - 5 V a poškození platformy Arduino Mega ADK.

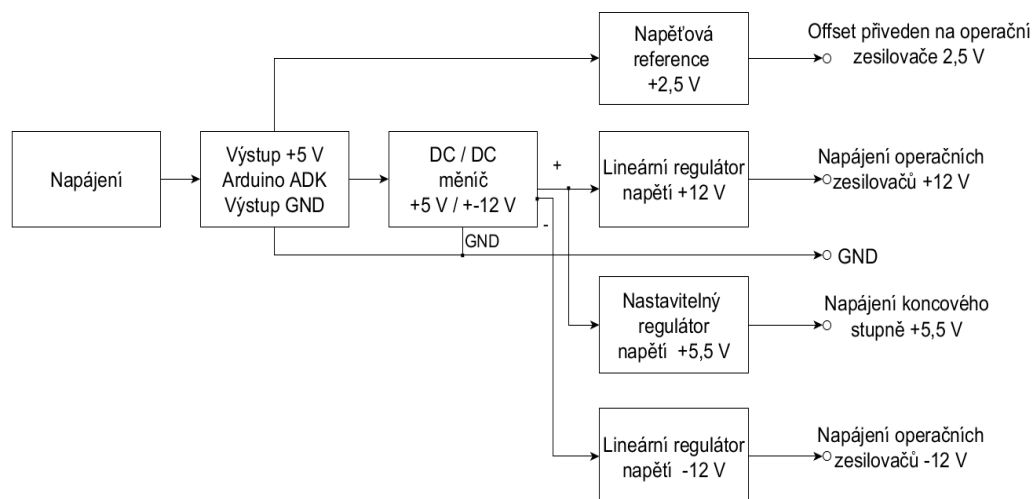
Koncový stupeň je složen ze čtveřice OZ OPA350. Jedná se o speciální vysokorychlostní, nesymetricky napájené OZ tzv. *rail-to-rail*<sup>2</sup>. Výstupní napětí je v rozmezí 10 mV od napájecího. Jsou zapojeny jako napětové sledovače na výstup každého IO2A-C. napájení je zajištěno 5V více kapitola 3.3.4. Je-li na jejich vstup přivedeno napětí záporné, tzn. výstup neinevrtujících sumačních OZ je záporný, omezí tuto hodnotu na 0 V. Naopak při hodnotě převyšující povolený rozsah dojde k jejich saturaci a výstupní napětí je omezeno na 5V. Tím jsou analogové vstupy platformy Arduino Mega ADK chráněny před poškozením.[16]

### 3.3.4 Napájení

Blokové schéma napájení obvodu, který je předřazen platformě Arduino Mega ADK je na obr. 3.8

- **Napájení** může být u platformy Arduino Mega ADK realizováno přes USB nebo pomocí externího zdroje napětí. Zvolena a testována byla možnost USB kabelu, protože je pomocí něj zajištěn jak přenos dat do PC, tak napájení.
- **Napětová reference** je napájena z výstupního 5 V pinu platformy Arduino Mega ADK. Jedná se o LM336-2.5, která byla vybrána díky hodnotě a přesnosti výstupního napětí  $2,5$  V  $\pm 1\%$  a vysoké tepelné stabilitě. Teplotní koeficient lze ovlivňovat pomocí trimru. Více kompletní schéma zapojení A. Přesné

<sup>2</sup>Rail-to-rail značí, že výstupní dynamický rozsah je velmi blízký napájecímu napětí



Obr. 3.8: Blokové schéma obvodu napájení

výstupní napětí je přiváděno na neinvertující vstup sumačních zesilovačů (IO2A - IO2D) a zajišťuje tak offset signálu.[16]

- **DC/DC měnič** byl vybírán s důrazem na to, aby jeho přepínací frekvence byla jiná než vzorkovací frekvence AD převodníku Arduina Mega ADK a nedocházelo tak ke vzájemnému ovlivňování.
- **+5V /  $\pm 12$  V** Vstupem do DC/DC měniče je výstupní 5 V pin platformy Arduino Mega ADK. Výstupem je napětí  $\pm 12$  V, sloužící jako symetrické napájení IO1A - IO1C a IO2A - IO2D. Lze použít např. AM1D-0512DZ s frekvencí spínání 125 kHz a přesností  $\pm 5\%$ .
- **Lineární regulátor napětí +12 V** slouží k regulaci výstupního napětí DC/DC měniče jeho výkyvech. Použit může být např. 78L12. Jelikož je součástí vnitřní konstrukce vybraného DC/DC měniče regulátor napětí, jehož přesnost je stejná s v uvedenou referencí ( $\pm 5\%$ ), není za jeho výstup nutné přidávat další. Navíc by kvůli úbytku napětí na regulátoru 1,7 V musel být použit DC / DC měnič s vyšším výstupním napětím, alespoň 14 V. Z těchto důvodů nebyl regulátor při konstrukci použit.
- **Lineární Regulátor napětí -12 V** jako je např. 79L12, taktéž nebyl při konstrukci díky stabilizaci napětí DC/DC měničem.
- **Nastavitelný regulátor napětí +5,5 V** slouží k napájení operačních zesilovačů OPA350 koncového stupně. Jedná se o LM317.[8]

### 3.3.5 Proudový odběr

Maximální proudový odběr navrhovaného zapojení je vypočítán jako součet maximální spotřeby všech součástek, které byly použity v zapojení:

- TL084,  $I_{max} = 1,4 \text{ mA} \cdot 7 = 9,8 \text{ mA}$
- AM1D-0512,  $I_{max} = 149,5 \text{ mA}$
- OPA350,  $I_{max} = 8,5 \text{ mA} \cdot 4 = 34 \text{ mA}$
- LM336-2,5,  $I_{max} = 10 \text{ mA}$
- LM317,  $I_{max} = 12 \text{ mA}$

Celkový proudový odběr obvodu je potom:  $I_{max} = 215,3 \text{ mA}$ . Vzhledem k tomu, že USB sběrnice dodává proud 500 mA, z kterých 40 mA spotřebuje samo Arduino Mega ADK, není třeba žádné externí napájení.

## 4 NAMĚŘENÉ PARAMETRY A PRŮBĚHY

### 4.1 Rozsah vstupního napětí

Analogové piny platformy Arduino Mega ADK, dokáží pracovat se vstupním signálem o velikosti 0 V - 5 V. Přeražený hardware upravuje přičtením offsetu vstupní signál, který je tak možné měřit v rozsahu  $\pm 5$  V. Pro měření signálu malých amplitud lze volit z trojce zesílení, více 3.3.2. Rozšiřujícím prvkem je zapojení, které zvětšuje vstupní rozsah na  $\pm 10$  V při zesílení  $A = 1$ .

### 4.2 Počet kanálů

Díky tomu, že platforma Arduino Mega ADK obsahuje pouze jeden 10bitový A/D převodník s postupnou aproximací, může být v daném časovém okamžiku převáděn vždy pouze jeden analogový kanál. Při čtení více, např. dvou kanálů je nutno použití multiplexeru. Nejprve se vzorkuje první kanál, následně dochází pomocí multiplexeru k přepnutí kanálu připojeného k A/D převodníku a vzorkování druhého kanálu. Proces přepínání mezi prvním a druhým kanálem se neustále opakuje.

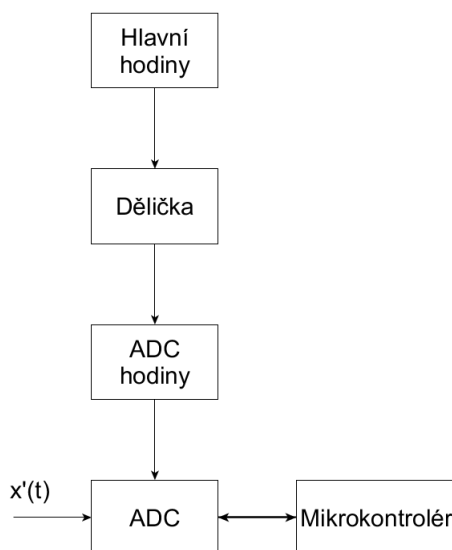
Návrh a měření bylo realizováno na nepájivém poli pro jeden kanál. Pokud bychom požadovali kanálů více, lze pro každý další kanál duplikovat kompletní zapojení uvedené v příloze A. Maximální počet kanálů je omezen počtem analogových pinů platformy Arduino Mega ADK na 4.

### 4.3 Vzorkovací frekvence

Pro určení hodnoty vzorkovací frekvence je důležitý způsob, jakým je u Arduina Mega ADK generována pracovní taktovací frekvence A/D převodníku z krystalového oscilátoru. Tento děj znázorňuje blokové schéma na obr.4.1.

Blokové schéma generování vzorkovací frekvence A/D převodníku vychází z původního obecného blokového schéma navrhovaného osciloskopu 3.1. Vstupem je analogový signál po zesílení a filtraci  $x'(t)$ .

- **Hlavní hodiny** generují taktovací frekvenci přímo z 16 MHz krystalového oscilátoru.
- **Dělička** generuje přijatelnou taktovací frekvenci pro A/D převodník a to dělením výstupní frekvence z krystalového oscilátoru. Dělička může nabývat hodnoty 2, 4, 8, 16, 32, 64 nebo 128 [1]
- **ADC hodiny** určuje počet pracovních cyklů A/D převodníku potřebných pro konverzi jednoho vzorku z analogového signálu na digitální. Normální doba



Obr. 4.1: Blokové schéma generování vzorkovací frekvence A/D převodníku

konverze A/D převodníku je 13 ADC cyklů s výjimkou počáteční, která trvá 25 ADC cyklů. Výstupem bloku je tedy reálná vzorkovací frekvence A/D převodníku.

Platforma Arduina ADK je osázena mikrokontrolérem ATmega2560. Podle dokumentace k tomuto mikrokontroléru je obvod A/D převodníku s postupnou aproximací schopen pracovat, při zachování 10bitového rozlišení, s taktovací frekvencí v rozsahu 50 kHz - 200 kHz. Technicky je možné dosáhnout limitní pracovní taktovací frekvence až 1 MHz. Společně s rostoucí taktovací frekvencí nicméně klesá i rozlišovací schopnost A/D převodníku. ADC modul obsahuje děličku, která generuje přijatelnou taktovací frekvenci pro ADC z krystalového oscilátoru. Podle nastavení příslušných bitů v registru ADPS nabývá dělička hodnoty 2, 4, 8, 16, 32, 64 a 128. Hodnoty jaké je třeba nastavit na bity registru ADPS pro výběr hodnoty děličky jsou uvedeny v tab.4.1.

Jelikož krystalový oscilátor platformy Arduino ADK pracuje s oscilační frekvencí 16 MHz, bude výstupní taktovací frekvence děličky následující:

- **Dělička 2:**  $16 \cdot 10^6 / 2 = 8 \cdot 10^6$  Hz
- **Dělička 4:**  $16 \cdot 10^6 / 4 = 4 \cdot 10^6$  Hz
- **Dělička 8:**  $16 \cdot 10^6 / 8 = 2 \cdot 10^6$  Hz
- **Dělička 16:**  $16 \cdot 10^6 / 16 = 1 \cdot 10^6$  Hz
- **Dělička 32:**  $16 \cdot 10^6 / 32 = 500 \cdot 10^3$  Hz
- **Dělička 64:**  $16 \cdot 10^6 / 64 = 250 \cdot 10^3$  Hz
- **Dělička 128:**  $16 \cdot 10^6 / 128 = 125 \cdot 10^3$  Hz

Tab. 4.1: Nastavení hodnoty dělicího poměru [1]

ADPS2	ADPS1	ADPS0	Dělicí poměr
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Pro zachování 10bitového rozlišení musí být taktovací frekvence v rozsahu 50 kHz - 200 kHz. Lze tedy použít pouze děličku 128 a dosáhnout tak taktovací frekvence A/D převodníku 125 kHz. Normální doba konverze A/D převodníku je 13 ADC cyklů s výjimkou počáteční, která trvá 25 ADC cyklů. Výsledná vzorkovací frekvence při zachování 10bitového rozlišení je tedy:

$$f_{vzMax} = \frac{16 \cdot 10^6}{128 \cdot 13} = 9,6 \cdot 10^3 \text{ Hz} \quad (4.1)$$

Takto vypočtená hodnota vzorkovací frekvence je pouze teoretická a nelze ji reálně dosáhnout. Nebere v úvahu další procesy vykonávané při běhu programu. Doba, kterou trvá převod hodnoty analogové na hodnotu digitální, je díky ostatním procesům v programu akviziční části 680  $\mu\text{s}$ . Tato hodnota byla vypočtena jako medián z měření času konverze tisíce vzorků. Vzorkovací frekvence je tedy:

$$f_{vz} = \frac{1}{680 \cdot 10^{-6}} = 1471 \text{ Hz}. \quad (4.2)$$

Pokles hodnoty vzorkovací frekvence je způsoben vykonáváním různých procesů cyklu jako je například: přepočítání naměřených hodnot a kontrola přicházejících dat z aplikace zobrazení (v jiném případě by uživatel nemohl měnit průběh akvizice dat). Největší podíl na pokles hodnoty vzorkovací frekvence má kontinuální odesílání naměřených hodnot po sériovém rozhraní vzorek po vzorku. To lze obejít ukládáním naměřených hodnot do bufferu a následným jednorázovým odesláním těchto hodnot vyprázdněním bufferu, nebo použitím jiné metody komunikace uvedené v podkapitole 3.2.3. Bez použití externího paměťového modulu je velikost paměti SRAM platformy Arduino Mega ADK 8 MB. Lze tedy uložit pouze několik stovek vzorků. Při tomto způsobu se už ale nejedná o kontinuální měření dat a v zobrazovaném průběhu by byly jasně viditelné skokové přechody kdy je signál vzorkován a kdy



jsou naměřené hodnoty pouze odesílány. Z tohoto důvodu byl zvolen kontinuální přenos naměřených hodnot.

## 4.4 Horní mezní frekvence

Hodota horní mezní frekvence je omezena vzorkovacím teorémem. Pro  $f_{vz} = 1471$  Hz bude horní mezní kmitočet:

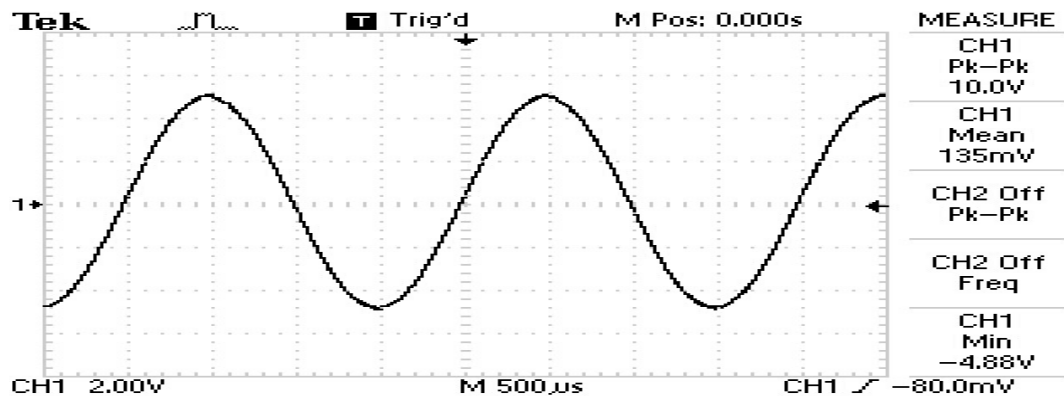
$$f_{Max} < \frac{1471}{2} < 735,5 \text{ Hz} \quad (4.3)$$

Frekvenční pásmo, které může být osciloskopem zobrazeno je 0 Hz - 735,5 Hz.

## 4.5 Naměřené průběhy

Vytvořený osciloskop byl testován pomocí generátoru funkcí FG-8002. Pro kontrolu parametrů výstupního signálu generátoru funkcí a signálu za koncovým stupněm, který vstupuje do platformy Arduino Mega ADK byl použit laboratorní osciloskop TDS 1012B. Jelikož je vzorkovací frekvence  $f_{vz} = 1471$  Hz, byl použit signál s následujícími parametry:  $U_{PK-PK} = 10$  V a  $f = 500$  Hz.

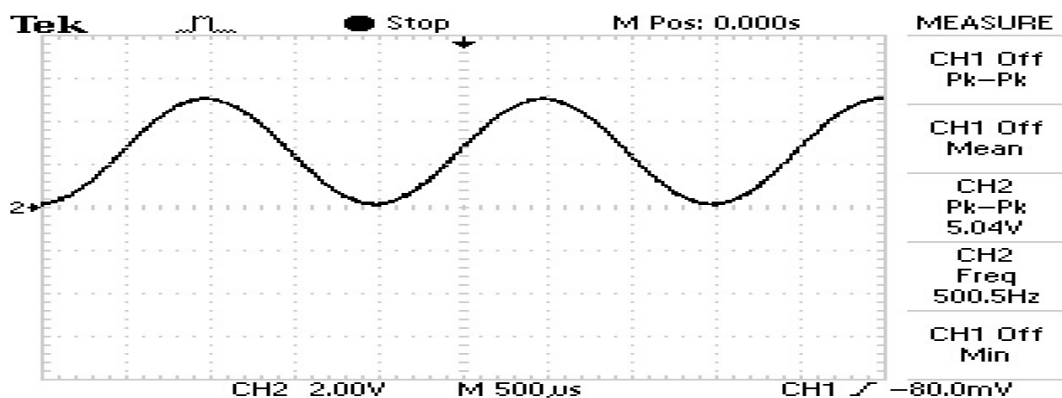
### 4.5.1 Sinusový průběh



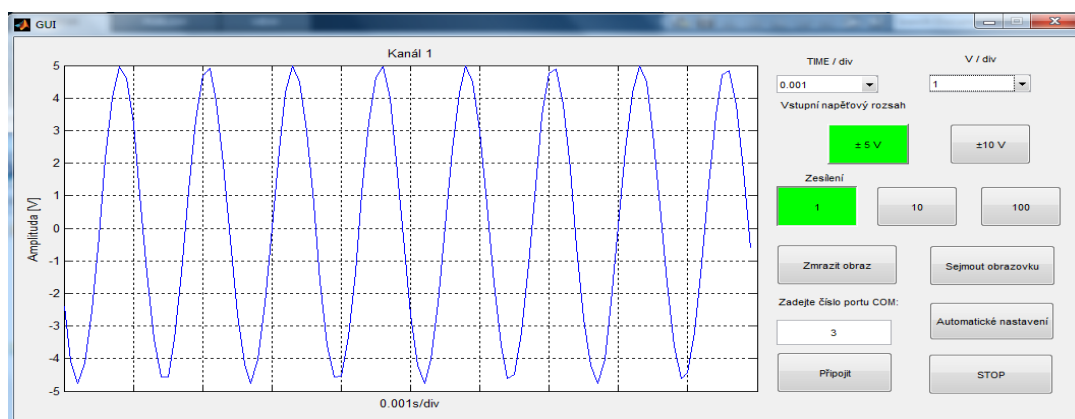
Obr. 4.2: Sinusový průběh z funkčního generátoru  $f = 500$  Hz,  $U_{PK-PK} = 10$  V

Na obr. 4.2 je zobrazen sinusový průběh z funkčního generátoru. Jedná se o vstupní signál do obvodu předřazenému platformě Arduino Mega ADK.

Původní měřený signál je hardwarovou částí vytvořeného osciloskopu zeslaben a je k němu přičten offset. Průběh signálu na výstupu koncového stupně je na obr. 4.3.



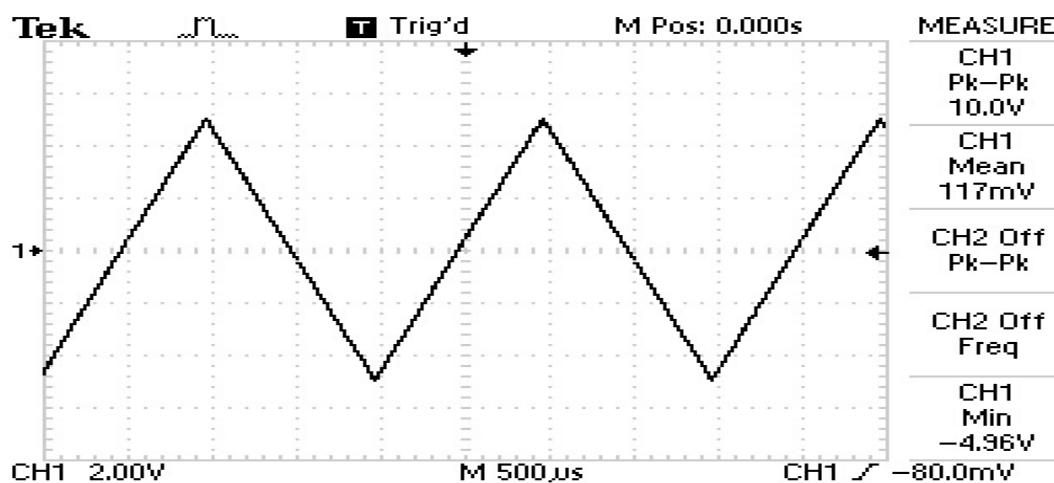
Obr. 4.3: Sinusový průběh naměřený za koncovým stupněm  $f = 500$  Hz,  $U_{PK-PK} = 5,04$  V



Obr. 4.4: Sinusový průběh naměřený vytvořeným osciloskopem  $f = 500$  Hz,  $U_{PK-PK} = 10$  V

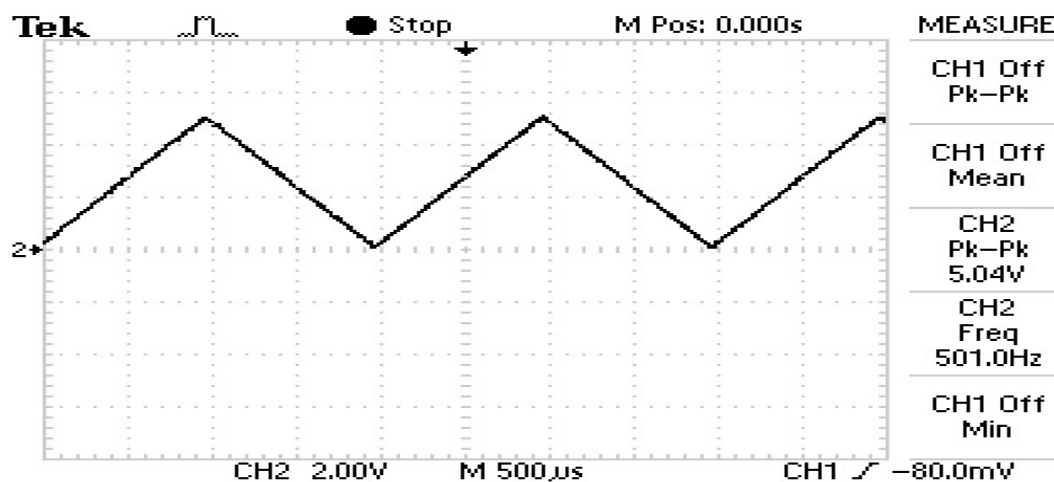
Naměřený sinusový průběh vytvořeným osciloskopem je na obr. 4.4. Zobrazený rozsah  $\pm 5$  V,  $A = 1$  odpovídá vstupnímu signálu. Vlivem nízké vzorkovací frekvence jsou přechody při maximální výchylce ostré v porovnání s laboratorním osciloskopem.

### 4.5.2 Trojúhelníkový průběh



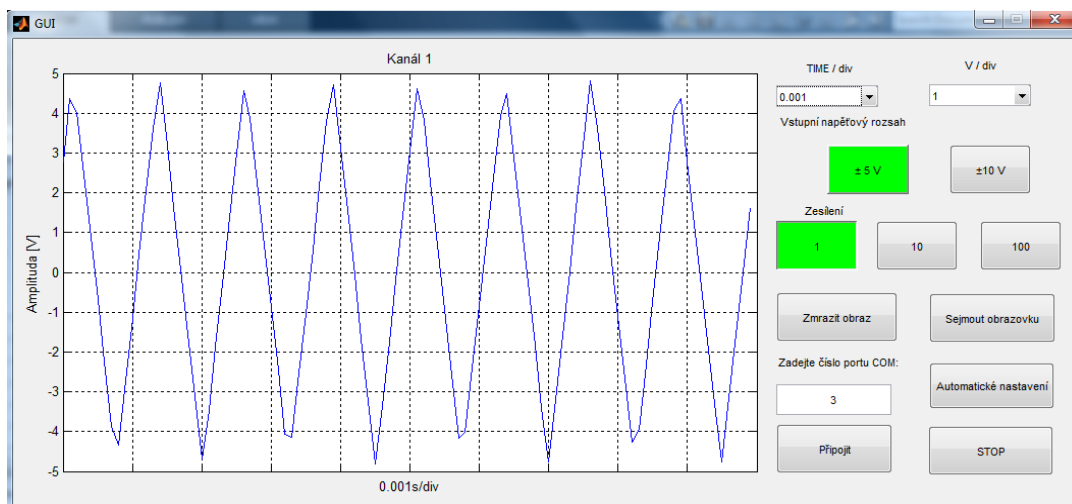
Obr. 4.5: Trojúhelníkový průběh z funkčního generátoru  $f = 500$  Hz,  $U_{PK-PK} = 10$  V

Trojúhelníkový průběh z funkčního generátoru, který je vstupním signálem do obvodu předřazenému platformě Arduino Mega ADK, je na obr.4.5.



Obr. 4.6: Trojúhelníkový průběh naměřený za koncovým stupněm  $f = 500$  Hz,  $U_{PK-PK} = 5,04$  V

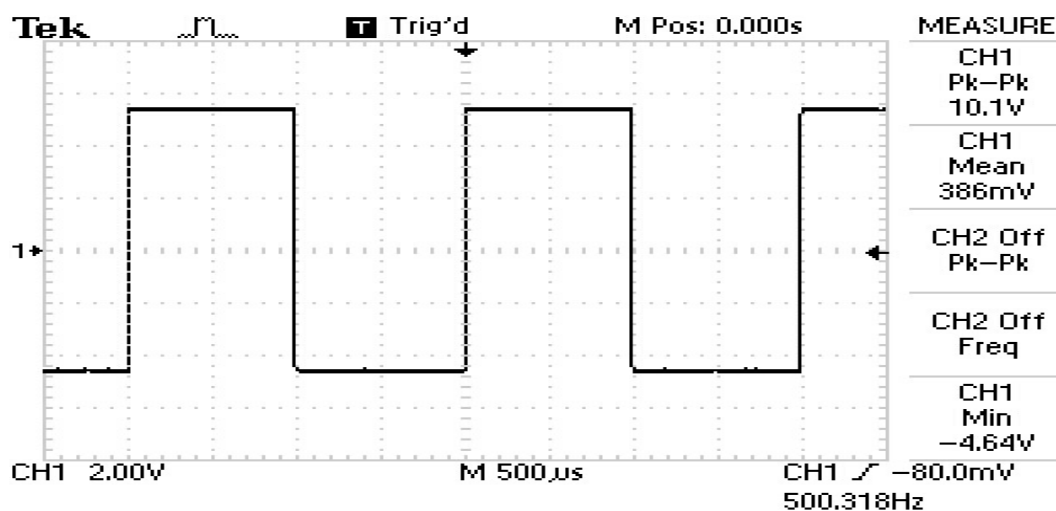
Průběh signálu na výstupu koncového stupně je na obr.4.6. Dochází ke stejným úpravám jako u sinusového průběhu.



Obr. 4.7: Trojúhelníkový průběh naměřený vytvořeným osciloskopem  $f = 500$  Hz,  $U_{PK-PK} = 10$  V

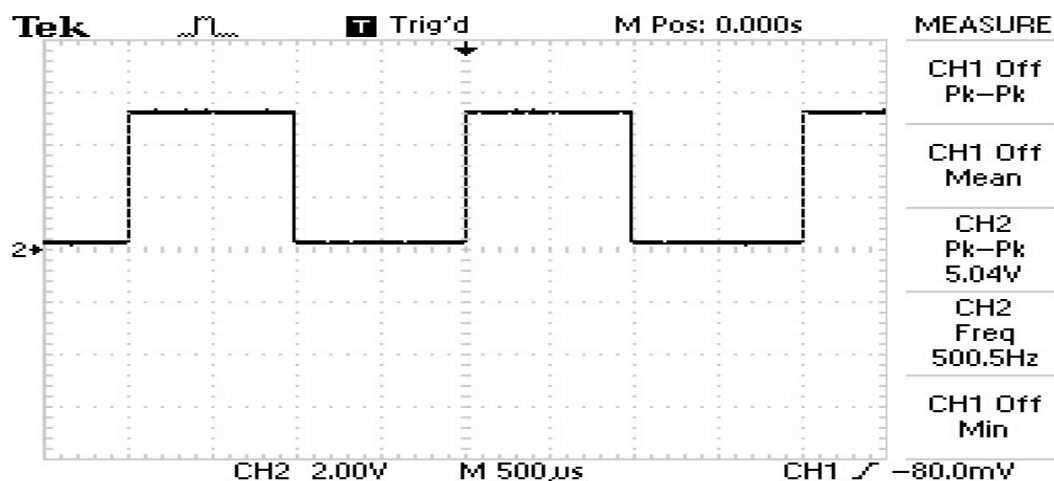
Zobrazovaný rozsah  $\pm 5$  V,  $A = 1$  odpovídá rozsahu vstupnímu signálu. Rychle se měnící vrcholy některých period trojúhelníkového vstupního signálu nejsou vlivem nízké vzorkovací frekvence zachyceny, což je vidět na obr. 4.7.

### 4.5.3 Obdelníkový průběh



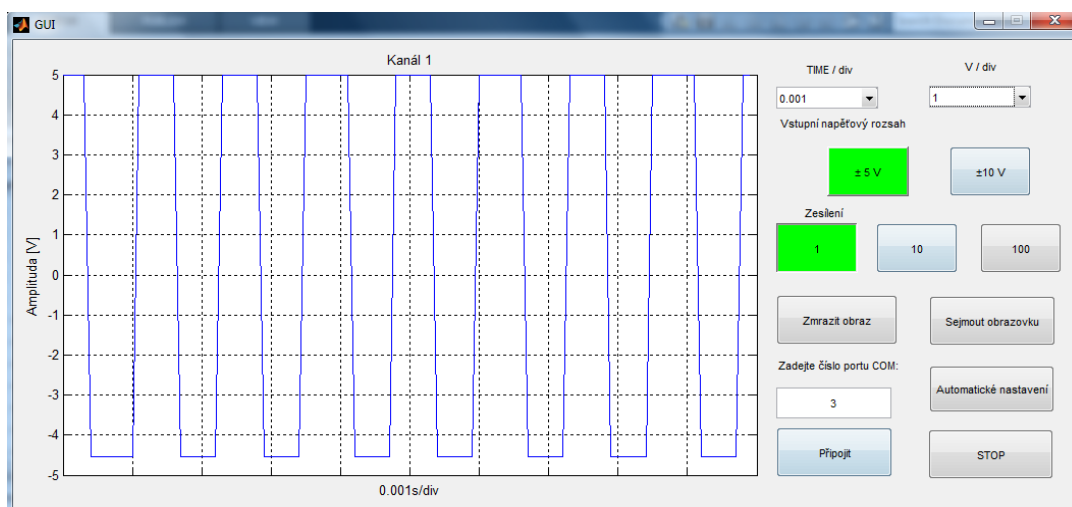
Obr. 4.8: Obdelníkový průběh z funkčního generátoru  $f = 500$  Hz,  $U_{PK-PK} = 10$  V

Vstupním signálem do obvodu předřazenému platformě Arduino Mega ADK, je obdelníkový průběh z funkčního generátoru obr. 4.8.



Obr. 4.9: Obdelníkový průběh naměřený za koncovým stupněm  $f = 500 \text{ Hz}$ ,  $U_{PK-PK} = 5 \text{ V}$

Na výstupu koncového stupně, je signál stejným způsobem jako u předchozích průběhů, posunut do oblasti 0 V - 5 V. Průběh signálu je na obr. 4.9.



Obr. 4.10: Obdelníkový průběh naměřený vytvořeným osciloskopem  $f = 500 \text{ Hz}$ ,  $U_{PK-PK} = 10 \text{ V}$

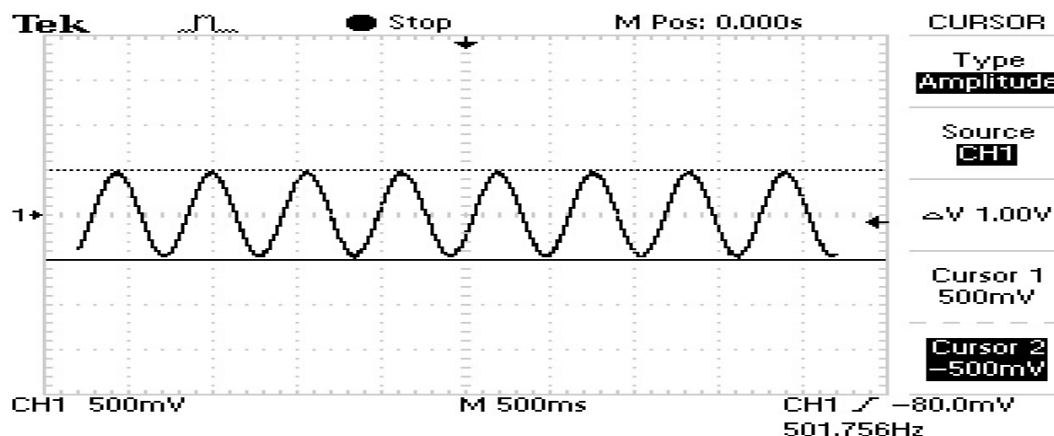
Obdelníkový průběh je díky rychlé změně vytvořeným osciloskopem nejhůře zobrazitelný. V náběžné i sestupné hraně obdelníkového signálu na obr. 4.10 je dobře vidět vzdálenost mezi jednotlivých vzorků.

#### 4.5.4 Měření zesílení vstupního signálu

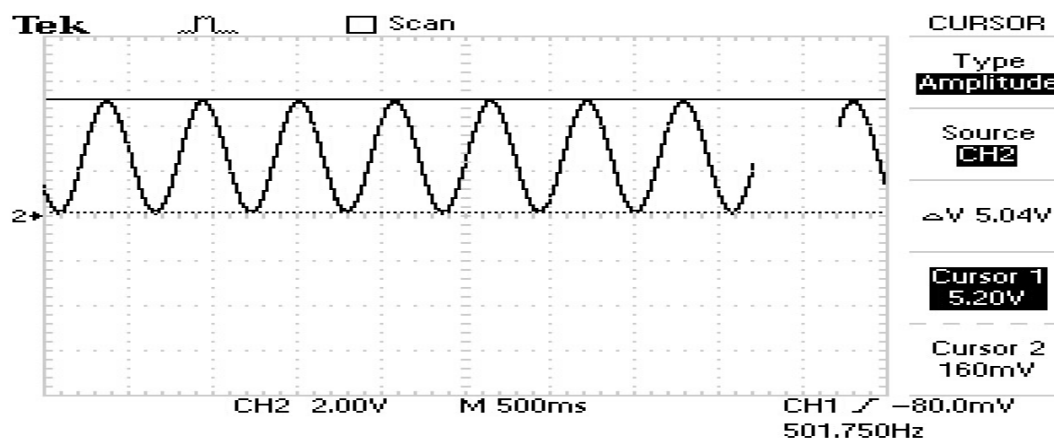
Uživatel má na výběr v rámci napětového vstupního rozsahu  $\pm 5$  V zvolit zesílení  $A_1 = 1$ ,  $A_2 = 10$  nebo  $A_3 = 100$ . Průběhy při zesílení  $A_1 = 1$ , jsou uvedeny v podkapitole 4.5.1.

##### Desetinásobné zesílení vstupního signálu

K měření desetinásobného zesílení byl použit signál s frekvencí  $f = 500$  Hz a Amplitudou  $U_{PK-PK} = 1$  V. Výstup signálu z funkčního generátoru je na obr. 4.11

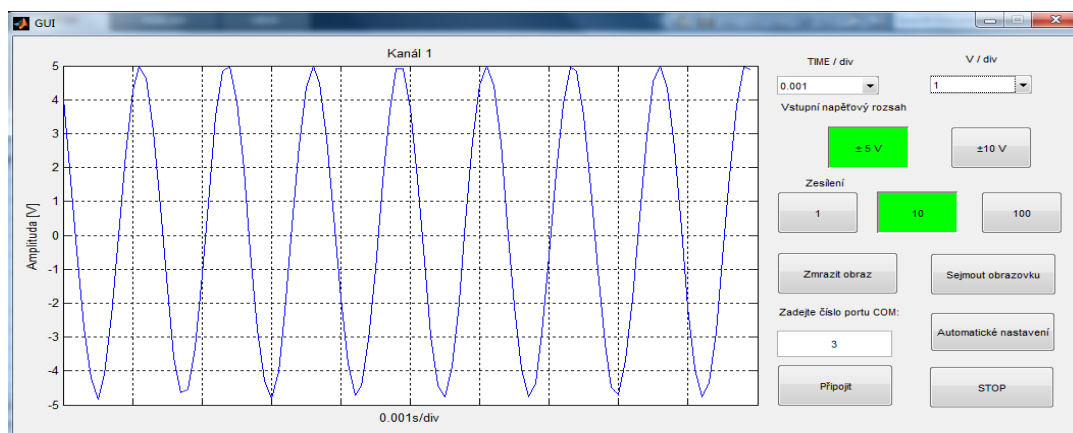


Obr. 4.11: Sinusový průběh z funkčního generátoru  $f = 500$  Hz,  $U_{PK-PK} = 1$  V



Obr. 4.12: Zesílený sinusový průběh naměřený za koncovým stupněm  $f = 500$  Hz,  $U_{PK-PK} = 5$  V

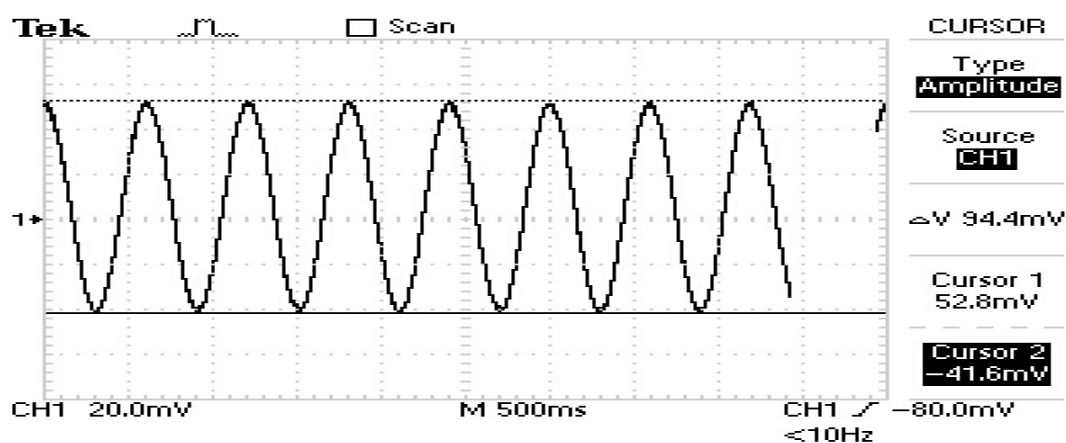
Zesílený sinusový průběh na obr.4.12 je po průchodu obvodem, který je předřazen platformě Arduino Mega ADK v maximálním rozsahu hodnot 0 V - 5 V. Tento průběh odpovídá desetinásobnému zesílení při vstupním napětí  $U_{PK-PK} = 1$  V.



Obr. 4.13: Zesílený sinusový průběh naměřený vytvořeným osciloskopem  $f = 500$  Hz,  $U_{PK-PK} = 10$  V

Průběh naměřený vytvořeným osciloskopem na obr.4.13 je po desetinásobném zesílení rozsahu  $U_{PK-PK} = 10$  V. K tomu, aby nedošlo k chybnému odečtení hodnoty je při stisku tlačítka A = 10 jeho aktivace indikována zelenou barvou.

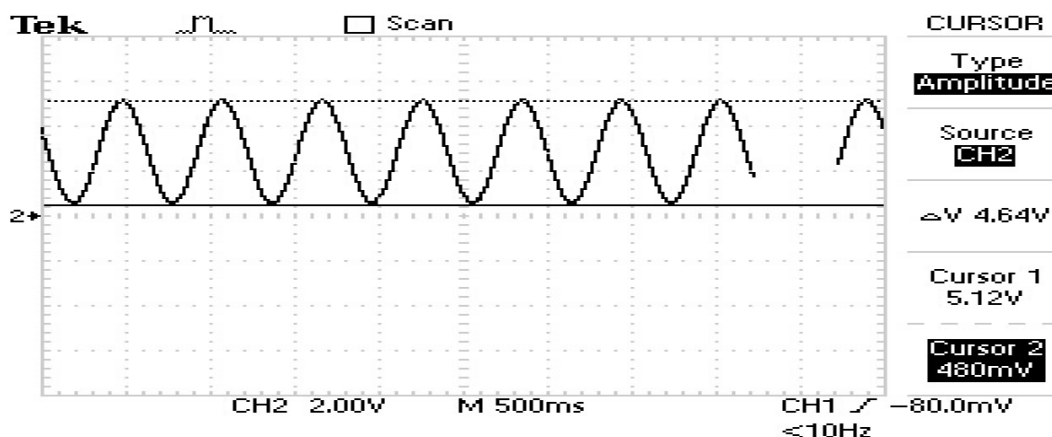
### Stonásobné zesílení vstupního signálu



Obr. 4.14: Sinusový průběh po průchodu odporovým děličem  $f = 500$  Hz,  $U_{PK-PK} = 100$  mV

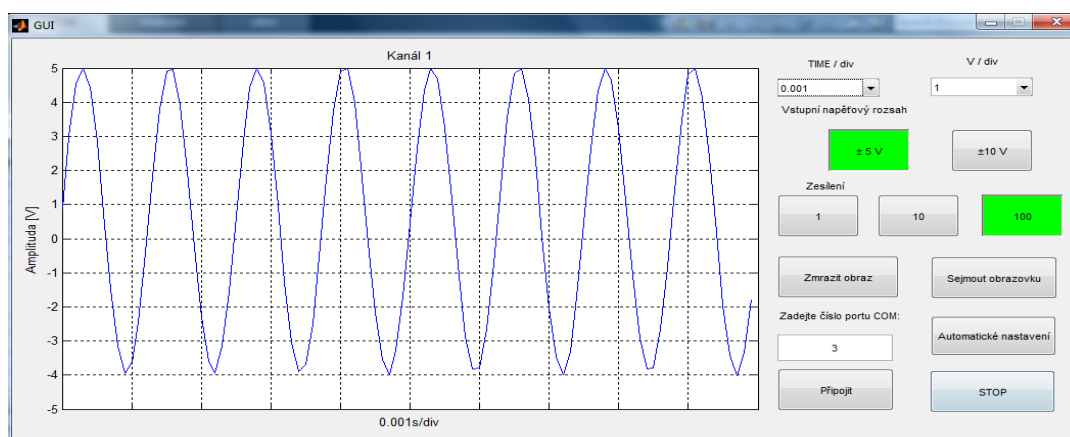
Výstupem funkčního generátoru při testování stonásobného zesílení byl signál s parametry:  $f = 500$  Hz,  $U_{PK-PK} = 100$  mV. Jelikož nejmenší amplituda výstupního

napětí funkčního generátoru FG-8002 je  $U_{PK-PK} = 200 \text{ mV}$ , byl pro měření použit odporový dělič s poměrem 1/2. Výstupní signál po průchodu odporovým děličem je na obr. 4.14



Obr. 4.15: Sinusový průběh naměřený za koncovým stupněm  $f = 500 \text{ Hz}$ ,  $U_{PK-PK} = 100 \text{ mV}$

Zesílený sinusový průběh naměřený po průchodu koncovým stupněm je na obr. 4.15. Stonásobné zesílení se již vlivem tolerance hodnot odporů nepodařilo přesněji nastavit. Dolní hranice rozsahu výstupního napětí je posunuta téměř o 500 mV.



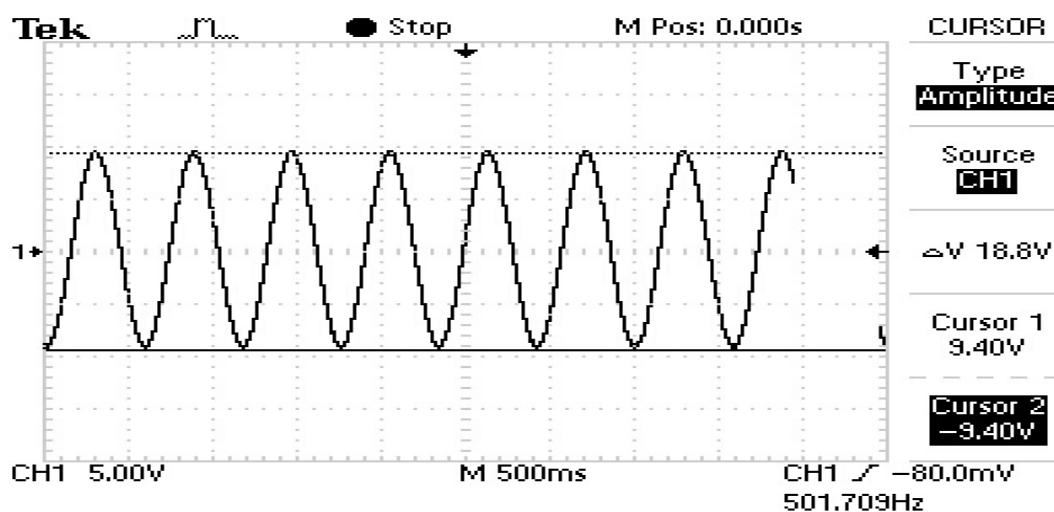
Obr. 4.16: Sinusový průběh naměřený vytvořeným osciloskopem  $f = 500 \text{ Hz}$ ,  $U_{PK-PK} = 100 \text{ mV}$

Jak již bylo řečeno výše, přesnou hodnotu stonásobného zesílení se nepodařilo realizovat vlivem tolerance odporů. Na obr. 4.16 je viditelný posun směrem do kladných hodnot, který koresponduje s posuvem výstupního napětí koncového stupně na obr. 4.15.



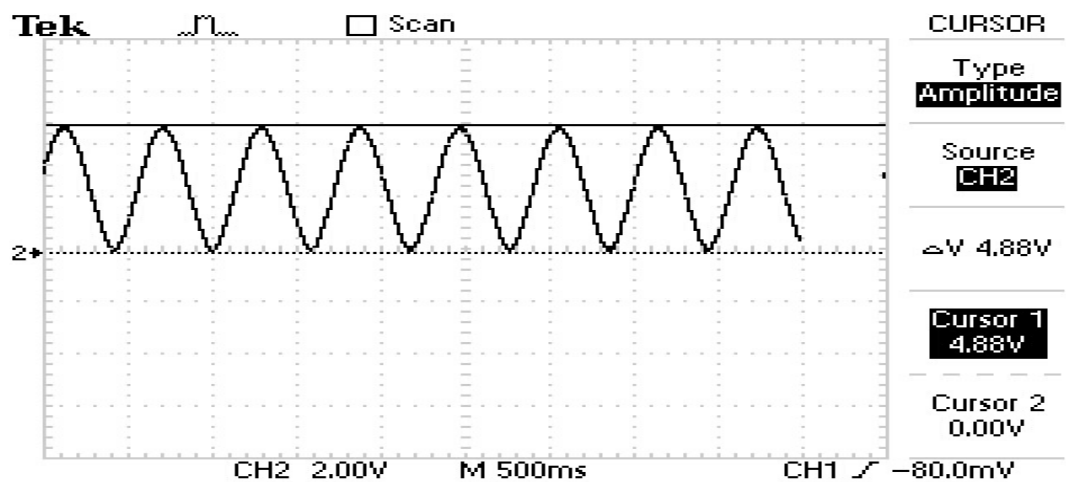
### 4.5.5 Měření rozsahu vstupního signálu

Ve všech předešlých případech byly průběhy měřeny v maximálním rozsahu vstupního napětí  $\pm 5$  V. Jako rozšiřující je použito zapojení zvětšující napěťový rozsahu na  $\pm 10$  V, o kterém je podrobněji pojednáno v podkapitole 3.3.2. Při měření se ukázalo, že i malé odchylky od vypočtených odporů, způsobují rozdílnou sumaci napětí. Na výstupu se proto při vstupním napětí na hranici rozsahu může objevit velmi malé záporné napětí. Proto je doporučeno měřit napětí v maximálním rozsahu  $\pm 9,4$  V. Výstupní signál z generátoru je na obr. 4.17.

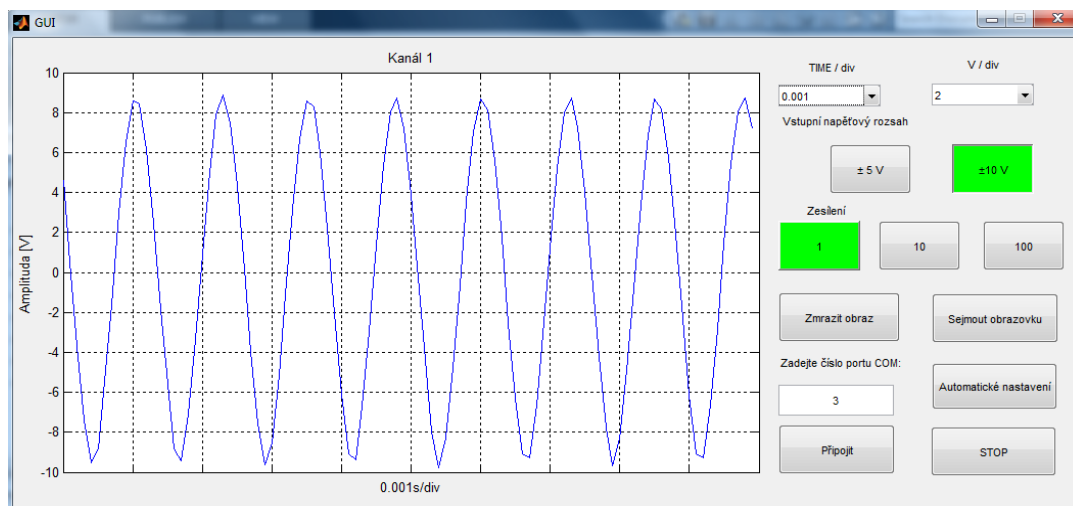


Obr. 4.17: Sinusový průběh z funkčního generátoru  $f = 500$  Hz,  $U_{PK-PK} = 18,8$  V

Signál naměřený po průchodu koncovým stupněm je na obr. 4.18. Maximální výchylka je od 0 V - 4,88 V. Díky tomu je amplituda naměřeného sinusového průběhu na obr. 4.19 v záporné půlce -9,5V a v kladné půlce pouze 8,5 V. Tato odchylka je způsobena opět tolerancí odporů, jelikož jsou v zapojení jedinným ovlivňujícím faktorem. Sumace s napěťovou referencí pak neproběhne přesně podle výpočtu, což se na výstupu projeví posunutím výstupního měřeného napětí koncového členu.



Obr. 4.18: Sinusový průběh naměřený za koncovým stupněm  $f = 500$  Hz,  $U_{PK-PK} = 5$  V



Obr. 4.19: Sinusový průběh naměřený vytvořeným osciloskopem  $f = 500$  Hz,  $U_{PK-PK} = 5$  V

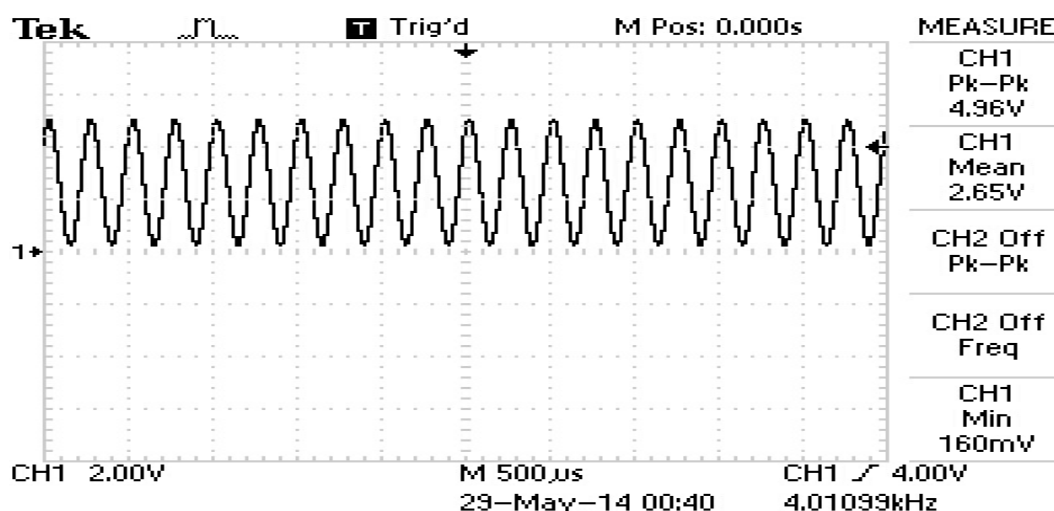
### 4.5.6 Zhodnocení naměřených výsledků

Hardwarová část osciloskopu, která se stará o úpravu signálu před samotným měřením pomocí analogových pinů platformy Arduino Mega ADK, funguje až na malé odchylky hodnot napětí správně. U procházejícího signálu nedochází k žádnému fázovému posunu. Při zachování povoleného maximálního rozsahu je výstupní napětí vždy v rozsahu 0 V - 5 V, s kterým pak Arduino Mega ADK může bezpečně dále pracovat.

Zesílení vstupního signálu se podařilo přesně nastavit pro hodnotu  $A = 10$ . Pro větší zesílení  $A = 100$ , dochází k maximálním odchylkám při měření vytvořeným osciloskopem 0,5V, což odpovídá odchylce 50 mV, způsobené při chybě sumace napětí.

Při měření s rozsahem vstupních napětí  $\pm 5$  V dostáváme poměrně přesné výsledky s odchylkami v řádech desetin V. Při použití rozšířeného rozsahu  $\pm 10$  V je již přesnost horší. Dochází k posunu napětí směrem do záporné části o 1 V.

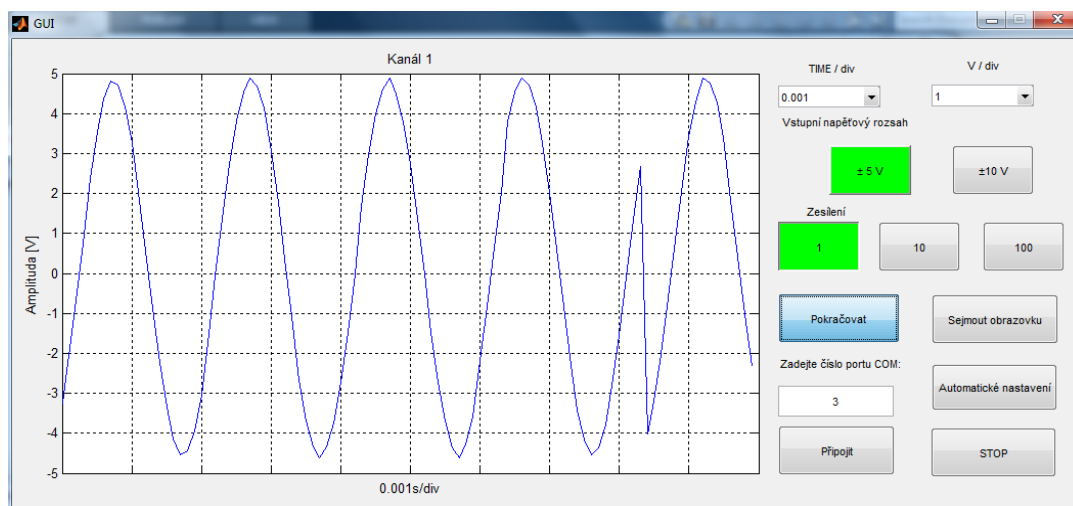
Výše popsané napěťové odchylky jsou způsobeny vlivem tolerance odporů. Jelikož jsou v obvodu mnohé velikosti odporů skládány, nedosáhne se často vypočítané hodnoty. To má za následek rozdílnou hodnotu zesílení nebo součet napětí v neinvertujícím sumačním zapojení zesilovačů.



Obr. 4.20: Sinusový průběh naměřený za koncovým stupněm  $f = 4000$  Hz,  $U_{PK-PK} = 5$  V

V podkapitole 4.3 je zmíněno, že jedním ze způsobů zvýšení vzorkovací frekvence je ukládání naměřených dat do paměti. Na obr. 4.20 je sinusový průběh za koncovým stupněm s frekvencí 4000 Hz. Jeho průběh naměřený vytvořeným osciloskopem byl vzorkován při nastavení děličky v programu akviziční části na hodnotu 16. Skok

na obr 4.21 je okamžik přechodu mezi odesílání dat, které jsou uloženy v bufferu a vzorkováním dat nových. Aby k takovým skokům nedocházelo byl zvolen přístup umožňující kontinuální měření.



Obr. 4.21: Sinusový průběh naměřený vytvořeným osciloskopem  $f = 4000$  Hz,  $U_{PK-PK} = 5$  V

## ZÁVĚR

Úvodní část bakalářské práce se zabývá teoretickým rozbořem nejvýznamnějších funkčních částí a vlastností číslicového osciloskopu. Důraz je kladen na převod spojitého vstupního analogového signálu na diskrétní a chyby, které tento proces ovlivňují. Dále jsou popsány hardwarové i softwarové části platformy Arduino, ve kterých jsou shrnuty nejdůležitější parametry a struktura programování integrovaného mikrokontroléru.

Praktická část bakalářské práce začíná návrhem a popisem jednotlivých funkčních bloků navrhovaného osciloskopu a přechází v jeho konkrétní softwarové vybavení. To se skládá z dvojice vzájemně komunikujících programů. Popis programu akviziční části hodnotí vývoj jeho různých verzí. Konečná verze programu kontinuálně vzorkuje vstupní signál s frekvencí  $f_{vz} = 1471$  Hz a přesností 10 bitů. Data jsou po sériové lince přenášena do GUI aplikace zobrazení, jejíž prostřednictvím může uživatel zasahovat do probíhající akvizice dat. GUI nabízí možnost měření střídavého i stejnosměrného signálu v rozsahu  $\pm 5$  V, s volitelným zesílením  $A_1 = 1$ ,  $A_2 = 10$ ,  $A_3 = 100$ . Rozšířením původního zadání je dále možnost měření signálu v rozsahu  $\pm 9,5$  V. Dále je možné: nastavit číslo portu COM, ke kterému je připojena platforma Arduino Mega ADK, sejmutí a uložení aktuální obrazovky a ruční změna napětového i časového měřítka. Implementována je taktéž automatická změna napětového měřítka, což u časového měřítka nebylo možné.

Hardware osciloskopu, který je předřazený platformě Arduino Mega ADK, vhodným způsobem upravuje vstupní měřený signál pro další zpracování. Tyto úpravy vstupního signálu jsou popsány, vysvětleny a odůvodněny společně s výpočtem hodnot vybraných součástek.

Testování správné funkce vytvořeného osciloskopu pomocí funkčního generátoru je náplní poslední části této práce. Naměřeny a popsány byly všechny průběhy, které bylo možno generovat.

V průběhu řešení práce se nepodařilo vyřešit spojení mezi systémem Android a platformou Arduino Mega ADK. Řešením by bylo spojení pomocí Bluetooth, což by vedlo k vyšší náročnosti bakalářské práce. Z toho důvodu bylo po dohodě s vedoucím práce zvoleno řešení propojením platformy s prostředím Matlab, které umožnilo splnit původní požadavky zadání.

# LITERATURA

- [1] ATMEL CORPORATION. Atmel: 8-bit AVR Microcontroller [online]. 2012 [cit. 2013-12-29]. Dostupné z: <http://www.atmel.com/Images/doc8161.pdf>
- [2] AGILENT TECHNOLOGIES. H TEST a.s. [online]. 2010 [cit. 2013-12-29]. Dostupné z: [http://www.htest.cz/download/1000\\_manual\\_CZ.pdf](http://www.htest.cz/download/1000_manual_CZ.pdf)
- [3] BARTUŠEK, Karel, Eva GESCHEIDTOVÁ, Radek KUBÁSEK, Jan MIKULKA, Jiří REZ a Miloslav STEINBAUER. Měření v elektrotechnice. 2., přeprac. a dopl. vyd. Brno: VUTUM, 2010, 212 s. ISBN 978-80-214-4160-6.
- [4] BANZI, Massimo, David CUARTIELLES, Tom IGOE, Gianluca MARTINO a D. MELLIS. ARDUINO. Arduino [online]. 2013 [cit. 2013-12-29]. Dostupné z: <http://arduino.cc/>
- [5] BANZI, Massimo. Getting started with Arduino. 1. vyd. Make: Books / O'Reilly, 2009. ISBN 978-059-6155-513.
- [6] DOSTÁL, Jiří. Operační zesilovače. 1. vyd. Praha: BEN, 2005, 504 s. ISBN 80-730-0049-0.
- [7] ĎAĎO, Stanislav a Josef VEDRAL. Číslicové měření: přístroje a metody. 1. vyd. Praha: ČVUT, 2006, 232 s. ISBN 80-010-2492-X.
- [8] FAIRCHILD. Semiconductor [online]. 2013 [cit. 2014-05-20]. Dostupné z: <http://www.fairchildsemi.com/>
- [9] HAVLÍK, Ladislav. Osciloskopy a jejich použití. Praha: nakladatelství Sdělovací technika, 2002, 254 s. ISBN 80-901-9368-4.
- [10] HÁZE, J.; VRBA, R.; FUJČÍK, L.; SAJDL, O. Vzájemný převod analogových a digitálních signálů - MEL608. Brno: N- PRESS, 2006.s. 1 ( s.)
- [11] MONK, Simon. Programming Arduino: getting started with sketches. New York: McGraw-Hill, c2012, xiv, 162 p. ISBN 978-007-1784-221.
- [12] OXER, Jonathan. Practical Arduino: cool projects for open source hardware. Berkeley, CA: Apress, c2009, xx, 423 p. ISBN 978-1-4302-2477-8.
- [13] PUNČOCHÁŘ, Josef. Operační zesilovače v elektronice. 4., dopl. vyd. Praha: BEN, 1999, 494 s. ISBN 80-860-5637-6.

- [14] SEDLÁČEK, Miloš. Zpracování signálů v měřicí technice. Vyd. 2. Praha: České vysoké učení technické, 1998, 234 s.
- [15] SEIBT, Artur. Osciloskopy od A do Z: technika obvodů, měřicí praxe, údržba. 1. čes. vyd. Ostrava: HEL, 2000, 255 s. ISBN 80-86167-11-9.
- [16] TEXAS INSTRUMENTS. TI Products: Semiconductors [online]. 1995 [cit. 2014-05-20]. Dostupné z: <http://www.ti.com/general/docs/prod.tsp>

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

A/D analogově-digitální převodník

ADC analogově-digitální převodník (z angl. Analog-to-digital converter)

OZ operační zesilovač

$f_{vz}$  vzorkovací kmitočet

S/H vzorkovač s pamětí (z angl. sample-hold)

I/H sledovač s pamětí (z angl. track-hold)

LSB nejméně významný bit (z angl. least significant bit)

MSB nejvíce významný bit (z angl. most significant bit)

I/O vstup/výstup (z angl. input/output)

USB univerzální sériová sběrnice (z angl. universal serial bus)

ADK vývojová kit (z angl. accessory development kit)

IDE integrované vývojové prostředí (z angl. integrated development environment)

PWM pulzně šířková modulace (z angl. pulse-width modulation)

ICSP metoda programování mikrokontrolerů (z angl. in circuit serial programming)

USART Synchronní / asynchronní sériové rozhraní  
(z angl. universal synchronous / asynchronous receiver and transmitter)

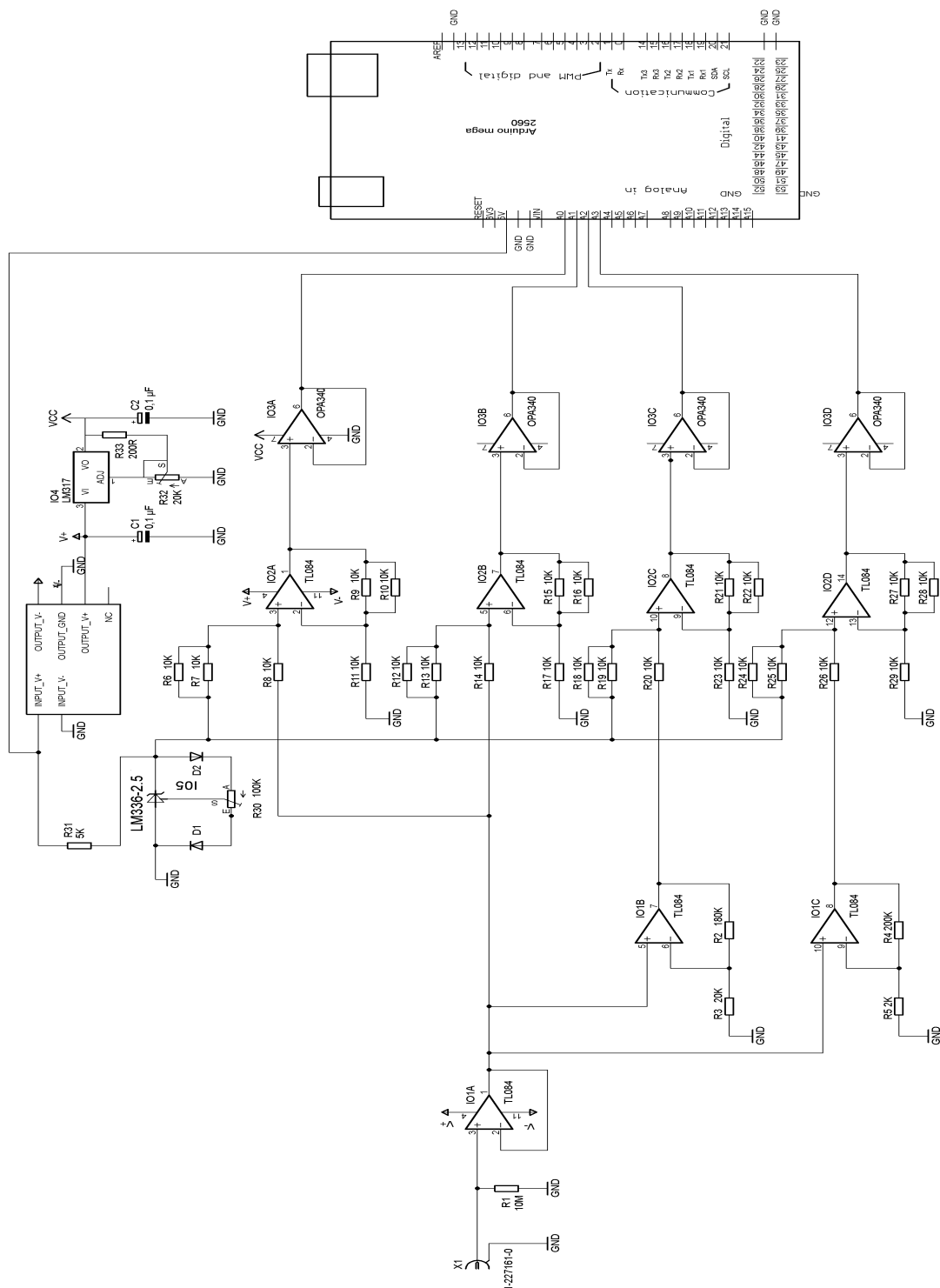
GUI Grafické uživatelské rozhraní  
(z angl. Graphical User Interface)



## SEZNAM PŘÍLOH

A	Kompletní schéma zapojení	66
B	Seznam použitých součástek	67

# A KOMPLETNÍ SCHÉMA ZAPOJENÍ



## B SEZNAM POUŽITÝCH SOUČÁSTEK

Počet	Označení	Hodnota
1	R1	10M
1	R2	180K
1	R3	20K
1	R4	200K
1	R5	2K
23	R6 - R29	50K
1	R30	100K
1	R31	5K
1	R32	20K TRIM
1	R33	200R
1	R4	200K
2	C1, C2	0,1 $\mu$
2	D1, D2	DIODE-DO35-10
2	IO1, IO2	TL084
1	IO3	OPA350
1	IO4	LM317LZ
1	IO5	LM336-2.5
1	IO6	AM1D-0512DZ
1	X1	CON-COAX