

Přípravek pro ověření vlastností digitálních osciloskopů

Diplomová práce

Studijní program:

N2612 Elektrotechnika a informatika

Studijní obor:

Mechatronika

Autor práce:

Bc. Tomáš Starý

Vedoucí práce:

Ing. Lubomír Slavík, Ph.D.

Ústav mechatroniky a technické informatiky





Zadání diplomové práce

Přípravek pro ověření vlastností digitálních osciloskopů

Jméno a příjmení: **Bc. Tomáš Starý**
Osobní číslo: M20000188
Studijní program: N2612 Elektrotechnika a informatika
Studijní obor: Mechatronika
Zadávací katedra: Ústav mechatroniky a technické informatiky
Akademický rok: **2021/2022**

Zásady pro vypracování:

1. Prostudujte problematiku digitálních osciloskopů.
2. Realizujte přípravek s řídicím mikroprocesorem ARM, který umožní ověřit důležité vlastnosti digitálních osciloskopů (metody spouštění: hranou, intervalem, stavem na sériové a paralelní digitální sběrnici a dalších metod; minimální doba hrany signálu, segmentovaná paměť, filtrace signálu, režimy zobrazování apod.).
3. Navrhněte laboratorní úlohu s využitím navrženého přípravku.

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:
Jazyk práce:

dle potřeby dokumentace
40–50 stran
tištěná/elektronická
Čeština



Seznam odborné literatury:

- [1] HAVLÍK, Ladislav. *Osciloskopy a jejich použití*. Praha: Sdělovací technika, 2002. ISBN 80-901936-8-4.
- [2] Asadi, Farzin, Eguchi, Kei: *Electronic Measurements: A Practical Approach*. Morgan & Claypool, 2021. ISBN-13: 978-1636390925
- [3] Yiu, Joseph: *The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors*. Newnes, 2013. ISBN-13: 978-0124080829

Vedoucí práce:

Ing. Lubomír Slavík, Ph.D.
Ústav mechatroniky a technické informatiky

Datum zadání práce:

12. října 2021

Předpokládaný termín odevzdání:

16. května 2022

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

doc. Ing. Josef Černožorský, Ph.D.
vedoucí ústavu

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědom toho, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

14. května 2022

Bc. Tomáš Starý

Přípravek pro ověření vlastností digitálních osciloskopů

Abstrakt

Tato diplomová práce se zabývá návrhem přípravku, který slouží k výuce práce s osciloskopem a k testování osciloskopů samotných. Přípravek je schopen generovat signály s různými předem nastavitelnými poruchami, jako je šum nebo náhodní peaky, tak přenosy po komunikacích.

Řízení přípravku obstarává mikrokontrolér. Součástí práce je řešení, softwarové řešení v mikrokontroléru, hardwarový návrh plošného spoje a vytvoření krytu zařízení pomocí 3D tisku.

Klíčová slova: generování šumu, testování osciloskopu, výuka s osciloskopy

Tool for verification of digital oscilloscopes characteristics

Abstract

The thesis is focused on the design of the device, which can be used for learning how to use oscilloscopes and for testing oscilloscopes. The device can generate signals with predetermined anomalies, like noise and randomised peaks, as well as communication using two different communication protocols.

The device is controlled by a microcontroller. The thesis is also focused on recherche of microcontroller software, the design of the hardware part of the circuit board, and the design of the device cover.

Keywords: noise generating, oscilloscope testing, education with oscilloscope

Poděkování

Tímto bych rád poděkoval svému vedoucímu práce Ing. Lubomíru Slavíkovi, Ph.D. za všechny odborné rady, které mi v průběhu vytváření této práce poskytl, a také Bc. Vítu Vaněčkovi za konzultace ohledně softwarového řešení přípravku.

Obsah

Seznam zkratk	12
1 Úvod	14
2 Osciloskop	16
2.1 Digitální osciloskop	17
2.2 Režimy snímání osciloskopu	19
3 Mikrokontrolér	25
4 Rešerše funkcí	30
4.1 Signál se šumem	30
4.1.1 Příčina šumu	31
4.1.2 Redukce šumu v měření s osciloskopem	32
4.1.3 Generování šumu	33
4.2 Zákmit na hraně	39
4.2.1 Ošetření zákmitu na hraně	40
4.3 Peaký na signálu	42
4.4 Komunikace I ² C	43
4.4.1 Fyzická vrstva	44
4.4.2 Princip přenosu	44
4.5 Komunikace SPI	45
4.5.1 Porovnání s I ² C	47
4.5.2 Paměť EEPROM	48
4.6 Paralelní komunikace	49

4.6.1	16x2 LCD displej	50
5	Praktická implementace funkcí	53
5.1	Generování šumu	53
5.2	Vytváření zákmitu na hraně	61
5.3	Peak na signálu	63
5.4	Simulace I2C	66
5.4.1	Ukázka měření	67
5.5	Simulace SPI	68
5.6	Paralelní sběrnice	69
6	Návrh plošného spoje	72
7	Návrh krabičky	76
8	Návrh laboratorní úlohy	79
9	Závěr	81
	Použitá literatura	82
A	Přílohy	87

Seznam obrázků

2.1	Katodový osciloskop	17
2.2	Blokové schéma digitálního osciloskopu [4]	18
2.3	Digitální osciloskop [6]	19
2.4	Režim XY [8]	23
2.5	Využití segmentované paměti při měření CAN komunikace[10]	24
3.1	Chip stm32 [11]	25
3.2	Porovnání frekvencí mikrokontrolérů STM[13]	28
3.3	Vývojová deska NUCLEO-H723ZG[14]	29
4.1	Použití průměrovacího filtru	33
4.2	Grafy generátorů náhodných čísel	35
4.3	Zenerova dioda	37
4.4	Bipolární tranzistor	38
4.5	Vakuová dioda 5722 od výrobce Sylvania[20]	39
4.6	Odskoky přepínače[23]	40
4.7	Peaky na reálném snímači.[25]	43
4.8	Princip zapojení sběrnice I2C	44
4.9	Časový průběh logických úrovní na vodičích SDA a SCL	45
4.10	Komunikace SPI[26]	46
4.11	Paměť EEPROM	48
4.12	Realizace buňky paměti EEPROM pomocí tranzistoru[28]	49
4.13	LCD display s integrovaným obvodem HD44780	50

5.1	Vytvořený šum	59
5.2	Porovnání spekter šumů	59
5.3	Generované šumy	60
5.4	Porovnání průběhu po průměrovacím režimu	61
5.5	Zákmit na signálu	63
5.6	Peak na signálu	65
5.7	Provedení čipu TC74[34]	67
5.8	Měřená probíhající komunikace I2C	68
5.9	Pouzdro paměti AT25010B-SSHL-B [36]	69
5.10	Adresa symbolů v ovladači HD44780[37]	70
6.1	Schéma zapojení přípravku	73
6.2	Návrh plošného spoje	74
6.3	Osazený plošný spoj	75
6.4	Osazený plošný spoj v krabičce	75
7.1	3D tiskárna technologie FDM[39]	76
7.2	Kryt při strukturální analýze	77
7.3	Kryt na zařízení	78
7.4	Hotová krabička s přípravkem uvnitř	78

Seznam zkratek

AC	Alternating Current (Střídavý proud)
DC	Direct Current (Stejnoseměrný proud)
CRT	Cathode Ray Tube (Katodová trubice)
VZ	Vstupní Zesilovače
ADC	Analog to Digital Converter (Analogově digitální převodník)
DAC	Digital to Analog Converter (Digitálně analogový převodník)
ČP	Číslicová Paměť
LED	Light Emitting Diode (Elektroluminiscenční dioda)
LCD	Liquid Crystal Display (Displej z tekutých krystalů)
AFG	Arbitrary Function Generator (Generátor libovolných funkcí)
RISC	Reduced Instruction Set Computer (Procesory s redukovanou instrukční sadou)
HAL	Hardware Abstraction Layer (Vrstva abstrakce hardwaru)
ARM	Advanced RISC Machine (Architektury procesorů)
RAM	Random Access Memory (Paměť s náhodným přístupem)
LSB	Least Significant Bit (Nejméně významný bit)
DHR	Digital Hold Register (Digitální přídržný registr)
DOR	Digital Output Register (Digitální vystoupí registr)
CPU	Central Processing Unit (Centrální procesorová jednotka)
RMS	Root Mean Square (Efektivní hodnota)
RND	Random Number Generation (Generátor náhodných čísel)
PDF	Probability Density Function (Hustota rozdělení pravděpodobnosti)
PN	Positive Negative junction (pozitivní negativní přechod)
SPST	Single Pole Single Throw (Jednopolový dvojpohový prepínač)
I2C	Inter-Integrated Circuit (Multi-masterová počítačová sériová sběrnice)
SPI	Serial Peripheral Interface (Sériové periferní rozhraní)
FIFO	First In First Out (První dovnitř, první ven)

LTO	Linear Tape Open (Otevřený formát pro páskové paměti)
EEPROM	Electrically Erasable Programmable Read Only Memory (Elektricky vymazatelná paměť pouze pro čtení)
LTO	Linear Tape Open (Otevřený formát pro páskové produkty)
CML	Centrální Mozek Lidstva
SSD	Solid state drive (polovodičový disk)
CMOS	Complementary Metal Oxide Semiconductor (Technologie integrovaných obvodů)
ATA	Advanced Technology Attachment (Sběrnice pro připojování zařízení k uchovávání dat)
SCSI	Small Computer System Interface (Standardní rozhraní a sada příkazů pro výměnu dat)
PCI	Peripheral Component Interconnect (Počítačová sběrnice pro připojení periferií k základní desce)
DMA	Direct Memory Access (Přímý přístup do paměti)
RWCR	Read Write Control Register (Povolení čtení zápisu)
ASCII	American Standard Code for Information Interchange (Americký standardní kód pro výměnu informací)
USB	Universal Serial Bus (Univerzální sériová sběrnice)
GND	Ground (Uzemnění)
FDM	Fused Deposition Modeling (Modelování depozicí taveniny)
FFF	Fused Filament Fabrication (výroba ztaveného vlákna)
STL	STereoLitografie (Formát souboru stereolitografického programu)

1 Úvod

Moderní osciloskop je jedním z nejpozoruhodnějších přístrojů, které byly kdy zkonstruovány. Zdlouhavé a intenzivní vývojové úsilí vynaložené na tyto přístroje se snad vyrovná pouze fanatismu věnovanému měření času. Je poctou konstruktérům osciloskopů, že přístroje vyrobené před více než 25 lety stále stačí pro více než 90 % dnešních měření. Je ovšem třeba mít na paměti zpoždění, šum, dobu náběhu, šířku pásma, odporové a kapacitní zatížení, přechod mezi kanály, nelinearitu zatížení, spouštění, přesnost a další omezení. Vysokorychlostní lineární obvody vyžadují od testovacího zařízení hodně a lze ušetřit nespočet hodin, pokud jsou dobře známé vlastnosti použitých přístrojů. Velké množství času bylo ztraceno sledováním „problémů s obvody“, které jsou ve skutečnosti způsobeny nepochopeným, nesprávně použitým nebo nevyhovujícím měřicím zařízením. Výborná znalost osciloskopu je neocenitelná pro dosažení nejlepších možných výsledků. V mnoha případech je možné použít zdánlivě nedostatečné zařízení k dosažení dobrých výsledků, pokud jsou omezení zařízení dobře známá a respektovaná. Nastává však problém, jak si všechny tyto znalosti měřicích osciloskopů osvojit a jak se naučit hledat chyby v zařízeních, pokud nevíme, jak se konkrétní chyby na osciloskopu projevují a jakým způsobem je detekovat? Právě z toho důvodu si tato práce dává za úkol vývoj a výrobu zařízení, které dokáže v kontrolovaných podmínkách generovat signály obohacené o předem známé a jasně specifikované typy poruch, jako jsou například šumy, náhodné impulsy, zákmity na hranách a mnoho dalších jevů, které se mohou vyskytnout při měření signálu. Moderní osciloskopy již dokáží měřit nejen analogové signály, ale také digitální komunikace, zobrazit datové celky putující mezi zařízeními a rozeznat, zda se v nich vyskytuje chyba. Proto zařízení vytvářené v této práci dokáže i tyto komuni-

kace využívat a měnit jejich parametry či simulovat chybové stavy. Úkolem obsluhy pak je tyto chyby nalézt, přičemž se má možnost dozvědět, jak takovéto signály, popřípadě komunikace, vůbec vypadají a jak se na nich dané chyby projevují a dají určit.

2 Osciloskop

Osciloskop je laboratorní přístroj běžně používaný k zobrazení a analýze tvaru vlny elektronických signálů. Zařízení přitom kreslí graf okamžitého napětí signálu jako funkci času. Nejběžnější použití osciloskopu je pro zobrazování střídavého (AC) nebo pulzujícího stejnosměrného (DC) signálu s frekvencí od jednoho hertzu po stovky megahertzů. Špičkové osciloskopy mohou zobrazovat signály dokonce až ve stovkách gigahertzů. Zobrazení na displeji je pro možnost odečtení hodnot rozděleno horizontálně i vertikálně na dílky. Na vodorovné stupnici se zobrazuje zleva doprava čas. Vertikální stupnice slouží k zobrazování hodnot napětí, kde kladné hodnoty jdou od osy směrem nahoru a záporné dolů. [1]

Mezi nejstarší varianty osciloskopů můžeme zařadit katodový osciloskop obr.2.1. Pracuje na principu paprsku elektronů, které jsou emitovány zahřátou katodou, urychleny a následně zobrazeny na fluorescenčním stínítku. Mezi elektronovým dělem a fluorescenční clonou jsou dva páry kovových desek. Desky jsou přitom orientovány tak, aby každý pár umožňoval vychylování paprsku ve jednom směru. Kombinace proměnné výchylky paprsku z těchto desek umožňuje řízený pohyb po stínítku. V místě dopadu elektronů na fluorescenční stínítko je excitován fosfor a tím emitováno světlo. Tato přeměna elektronové energie na světlo umožňuje zobrazovat body nebo křivky na jinak zatemněné obrazovce. [2]



Obrázek 2.1: Katodový osciloskop

Modernější osciloskopy elektronicky replikují činnost katodových osciloskopů pomocí displeje z tekutých krystalů (LCD) podobným těm, které se nacházejí v většině počítačových monitorů. Nejsofistikovanější osciloskopy využívají ke zpracování a zobrazení křivek počítače. Tyto počítače mohou používat jakýkoli typ displeje, který můžeme nalézt v moderních noutboocích, obrazovkách či telefonech.

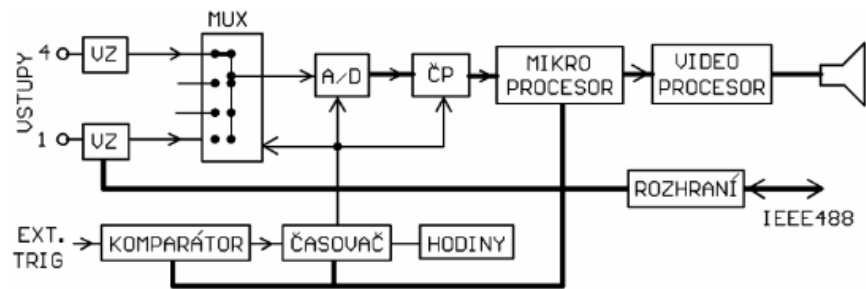
2.1 Digitální osciloskop

Digitální osciloskop, nebo jak je často označován digitální paměťový osciloskop, je komplexní elektronické zařízení složené z různých softwarových a elektronických hardwarových modulů. Tyto moduly spolupracují na zachycování, zpracování, zobrazení a ukládání dat. Data představují bitový zápis signálů, o které má operátor zájem. [3]

Ve své nejjednodušší podobě obsahuje digitální osciloskop šest prvků.

- Analogové vertikální vstupní zesilovače (VZ)
- Analogově-digitální převodník (A/D) a číslicovou paměť digitálních průběhů (ČP)

- Časovou základnu se spouštěčem a hodinovým modulem
- Obvody pro zobrazování a restrukturalizaci průběhu
- LED nebo LCD obrazovkou
- Zdroj energie



Obrázek 2.2: Blokové schéma digitálního osciloskopu [4]

Digitální osciloskopy obr.2.3 periodicky vzorkují časově proměnný analogový signál a ukládají ho do paměti i s informací o času záznamu. Pomocí vnitřních hodin rozdělují vstupní signály do samostatných časových bodů, kdy okamžité hodnoty amplitudy jsou v těchto bodech kvantovány osciloskopem. Výsledné digitální prezentace jsou následně uloženy v digitální paměti. Při předem stanoveném taktu se displej z paměti regeneruje a přitom zobrazuje naměřené hodnoty jako spojnicí bodů. Digitální osciloskopy poskytují výkonné funkce pro zobrazování a analýzu digitalizovaných dat z paměti. Některé z výhod digitálního osciloskopu, oproti analogovému osciloskopu, zahrnují schopnost ukládat digitální data do paměti pro pozdější prohlížení, přenos dat do počítače, generování papírové kopie nebo uložení na disk. Výhodou je také jeho schopnost okamžitě provádět měření digitálních dat. Další výhodou digitálních osciloskopů je možnost nastavení spouštěcí události, kdy mohou být digitální osciloskopy nastaveny tak, aby zobrazovaly průběhy, ve kterých dojde k spouštěcí události. Oproti tomu analogové osciloskopy je třeba spustit před zahájením sledování a po ukončení zase zastavit. Digitální osciloskop má také schopnost zkoumat digitalizované informace uložené v jeho paměti a provádět automatická

měření na základě zvolených parametrů uživatele, jako je výchylka napětí, frekvence a doba náběhu. Může také zobrazit podrobná zachycená data různými způsoby. Tato schopnost je umožněna přítomností většího počtu zachycených dat, než kolik je zobrazeno na obrazovce. Digitální osciloskopy také nabízí flexibilitu v podobě zobrazovací grafiky, např. dělení obrazovky na čtvrtiny nebo poloviny a vícekrokové programy pro zpracování dat. Digitální osciloskop je ideální pro zobrazování složitých křivek signálu, kde je třeba provádět výpočty a měření na konkrétních částech křivky. [5]



Obrázek 2.3: Digitální osciloskop [6]

2.2 Režimy snímání osciloskopu

Digitální osciloskopy se vyznačují tím, že umožňují uživateli volit mezi režimy záznamu, měnit délku záznamu, horizontální polohu a tvar vlny, či nastavovat funkce jako jsou: Fast Acq, Delay a XY Display.

Podstatou digitalizace signálu je, že změřené body definující tvar vlny představují vzorky. Analogově-digitální převodník (ADC), základní součást digitálního osciloskopu, určuje rychlost, jakou jsou vzorky odebrány. Převodník je regulován procesorem a externími hodinami (obdélníkový oscilátor). Doba mezi body vzorku se

nazývá "interval vzorku". Vzorkové hodnoty jsou nejprve uloženy do paměti a poté zobrazeny tak, aby vytvořili průběh vlny signálu.

Vzorkovací frekvence a frekvence vlny měřeného signálu se nemusí vždy shodovat. Důvodem je, že existuje více režimů akvizice, které ovlivňují způsob, kterým sekvenčně získané vzorkové body obsahují jednotlivé vlnové body. Režimy akvizice jsou: Sample, Peak Detect, High Resolution, Envelope a Average. [7]

Sample mode

V režimu Sample mode, nejzákladnějším typu snímání, osciloskop automaticky vytvoří tvar vlny uložení jednoho vzorkovacího bodu během každého intervalu snímaného signálu. Pro tuto metodu, kdy každé snímání odpovídá jednomu intervalu signál, musí mít měřená signál konstantní frekvenci.

Peak Detect mode

V režimu Peak-Detect se analyzují dva průběhy. Osciloskop uloží do své paměti minimální a maximální hodnoty, které tvoří dva odpovídající body průběhu. Protože režim Peak-Detect vyžaduje, aby ADC pracoval s rychlou vzorkovací frekvencí s pomalým nastavením časové základny, vede to k dlouhým interválům průběhu. Díky tomu ale můžeme zachytit rychlé změny signálu. V režimu Peak-Detect může osciloskop zobrazit rychlé přechodové jevy při pomalé rychlosti snímání, jako například když se krátké impulsy vyskytují daleko od sebe.

High Resolution mode

Režim vysokého rozlišení se podobá režimu Peak-Detect v tom, že ADC sampuluje rychleji, než je požadováno pro dané nastavení časové základny. Pokud se zvýší pozornost na typické stopy snímané osciloskopem, jako jsou například sinusové vlny vysílané z interního AFG, lze spatřit, že některé z nich vypadají ostřejší než jiné. Je to zapříčiněno mírou šumu, jenž obsahují. (Ve frekvenční oblasti se šum jeví jako kolísající hladina při nižší amplitudě než sledovaný signál. V časové oblasti se šum jeví jako ztlustění nebo malá nepravidelnost ve stopě.) Při vysokých frekvencích signál obsahuje více šumu což v režimu vysokého rozlišení zůstane zachováno. Při kombinaci s režimem průměrování,

dochází k efektivnějšímu potlačení šumu na nízkých frekvencích, avšak na rozdíl od režimu průměrování pracuje režim vysokého rozlišení i na neperiodickém signálu.

Envelope mode

Režim obálky se podobá režimu Peak-Detect, ale s rozdílem, že minima a maxima křivky jsou vzorkovány z různých akvizic, aby se následně v paměti vytvořila křivka, která v průběhu času vykazuje minimum a maximum mimo obálku. Režim Peak-Detect nejprve získá záznam a následně teprve použije tato surová data data k vytvoření obálkového průběhu. Při pohledu na režim obálky s počtem vrcholů nastaveným na nekonečno, bude stopa jednoduché sinusovky, generovaná z interního AFG, vykreslena jako tlustá čára o průměru dvou amplitud původního signálu. Toto ale není zobrazení šumu, je to statistická odchylka ve 2 000 křivkách z kterých se skládá zobrazovaná obálka.

Average mode

Average mode je užitečný při zobrazení signálu s velkým šumem, protože většina šumu, zejména bílého, je náhodná a při zprůměrování po sobě jdoucích křivek se automaticky ruší. Na rozdíl od opakujícího se signálu, který je zachován. Výhoda oproti omezení šířky pásma spočívá v tom, že nedochází ke snížení rozsahu frekvence měření.

Protože osciloskop může uložit pouze omezený počet vzorků, je obvykle považováno za žádoucí mít dlouhou délku záznamu. Avšak jsou i takové aplikace, kdy z důvodu docílení optimalizace výkonu může být délka záznamu záměrně zkrácena. Maximální délka záznamu pro konkrétní přístroj je uvedena ve specifikacích, ale uživatel ji pro citlivější měření může zkrátit.

Vzorkovací frekvence je určena délkou záznamu, dělenou časem snímání. Proto může být nutné zkrátit délku záznamu, kvůli paměťovým omezením v závislosti na frekvenci sledovaného signálu.

FastAcq mode

Účelem FastAcq je umožnit vysokorychlostní záznam tvaru vlny, což je užiteč-

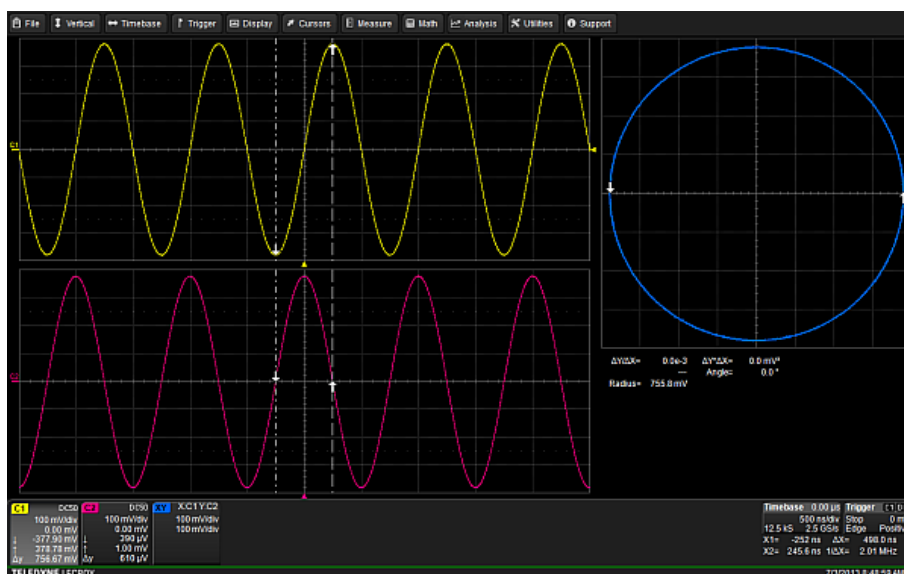
né při hledání náročněji zaznamenaných anomálií signálu. Snižuje mrtvou dobu mezi jednotlivými záznamy křivek, což umožňuje zachycení a zobrazení runtů a závad. Tento mód také zobrazuje charakteristiky tvaru vlny s intenzitou, která odpovídá počtu jejich výskytů. To je ekvivalentní zobrazení, jako na trojrozměrném grafu s třetí osou Z, který by vykreslovala množství výskytů, což by ale na plochém display nevypadalo dobře.

Wave Inspector

I poté, co původní signál již neexistuje, lze tvar vlny měřit a analyzovat v osciloskopu pomocí funkce Wave Inspector. Tato funkce dokáže měřit, hledat, testovat, přibližovat, posouvat a označovat data, která je třeba. Osciloskop lze navíc nastavit tak, aby se automaticky zastavil po dokončení záznamu nebo poté, co byla série záznamů převedena na obálku nebo průměrný průběh. Tato funkce se aktivuje stisknutím tlačítka Single Sweep, které se obvykle nachází u tlačítka Start/Stop poblíž tlačítka pro spuštění záznamu.

Režim XY

V režimu XY se digitalizovaný průběh nejprve skládá ze série bodů umístěných na kartézských souřadnicích obr.2.4 na osách X a Y. Každá z těchto teček je definována jednoznačně. Je nutné je ale propojit, aby vytvořily jasné zobrazení, kde jsou k dispozici informace nejen pro jednotlivé body, ale pro jakékoli místo na křivce. Aby k tomu došlo, a byla vytvořeno pouze jedno digitální zobrazení ekvivalentní původnímu analogovému vstupu, musí být splněn Nyquistův požadavek na vzorkování: Vzorkovací frekvence musí být minimálně dvojnásobkem nejvyšší frekvence obsažené v signálu. Signál musí být periodický, avšak může být i nesinusový, a tím pádem i dosti složitý. V takovémto případě může být každý periodický signál rozložen na řadu dokonalých sinusových signálů, které jsou celočíselnými násobky základní frekvence.



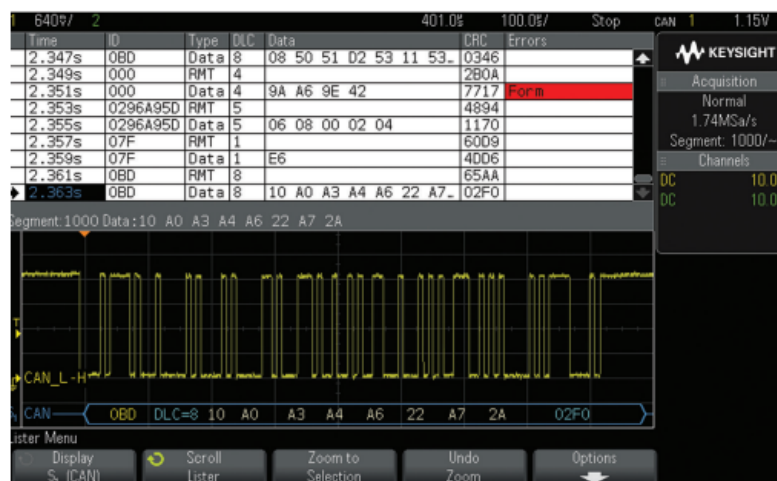
Obrázek 2.4: Režim XY [8]

Nejvyšší frekvence vzorkovaného signálu určuje minimální potřebnou vzorkovací frekvenci. Vzorkovací frekvence tedy nezávisí pouze na velikosti základní frekvence, ale také na složitosti vlny, která se objevuje na analogovém vstupu, a dále závisí i na šířce pásma osciloskopu. Přístroje s menší šířkou pásma, kdy se využívá omezení šířky pásma jako techniky pro potlačení vysokofrekvenčního šumu, často spíše než šum je eliminuje informace v signálu, která může být pro měření hodnotná. [9]

Segmentovaná paměť

Pokud signály, které je potřeba zachytit, mají relativně dlouhé doby nečinnosti mezi krátkými cykly zatížení nebo pulzní signálovou aktivitu, pak je možnost využití segmentované paměti. Segmentovaná paměť dokáže optimalizovat akviziční paměť osciloskopu, což umožní zachytit více selektivních detailů signálu s menším zaplněním paměti. Se segmentovanou pamětí je akviziční paměť osciloskopu rozdělena do několika menších paměťových segmentů. Díky tomu může osciloskop zachytit mnoho po sobě jdoucích průběhů jednoho obrazu s velmi krátkou dobou načítání, aniž by unikly důležité informace o signálu. Po provedení akvizice do segmentové paměti si lze snadno prohlédnout všech-

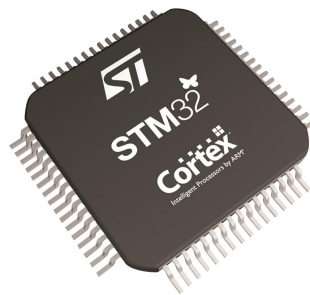
ny zachycené obrazy. Překrýt jednotlivé segmenty pro porovnání na displeji nebo rychle procházejte jednotlivé průběhy. Mezi běžné aplikace pro tento typ akvizice osciloskopu patří měření fyziky vysokých energií, laserových pulzů, radarových záblesků a paketové sériové sběrnice obr.2.5. Dokonce i v aplikacích, které ve skutečnosti nevyžadují segmentovanou akvizici paměti pro optimalizaci paměti, může použití zlepšit orientaci pro analýzu signálů, jako například u signálů s krátkým pracovním cyklem, u burstových signálů a sériově paketrovaných signálů. Některé osciloskopy poskytují segmentované paměťové akvizice současně na všech analogových kanálech nebo logických kanálech akvizice, a nebo také dokáží hardwarové sériové dekódování paketrovaných sériových dat pro každý zachycený segment průběhu.[10]



Obrázek 2.5: Využití segmentované paměti při měření CAN komunikace[10]

3 Mikrokontrolér

Při výběru mikrokontroléru bylo rozhodnuto pro mikrokontrolér z rodiny 32bitových mikrokontrolérů STM32 založených na procesoru ARM. Jedním z hlavních důvodů je jeho schopnost provádět vysokorychlostní operace díky řadě procesorů ARM Cortex. Vzhledem k tomu, že se jedná o RISC procesor, může pracovat a relativně vyšší rychlostí a provádět miliony instrukcí za sekundu. Dalšími podstatnými výhodami je velká rozšířenost, a tak i rozsáhlá podpora napříč programátorskou komunitou, možnost použití vývojových desek osazených těmito kontroléry, a díky tomu snadnější vývoj. V neposlední řadě přehledná dokumentace usnadňující programování a vlastní programovací prostředí s možností využití HAL.



Obrázek 3.1: Chip stm32 [11]

Při výběru konkrétní série a z této série určitého čipu byl kladen důraz na nejdůležitější vlastnost vytvářeného zařízení, kterou je generování signálu pro analýzu osciloskopem. Generovaný signál musí být co nejvíce věrnou podobou signálům běžně analyzovanými osciloskopem, a proto musí být i DAC co nejkvalitnější.

DAC převodník je elektronický obvod, který přijímá digitální číslo nebo hodnotu jako vstup a převádí je na analogové napětí, napětovou úroveň, která odpovídá

binárnímu číslu ve výstupním registru DAC. Výstupní napětí DAC se změní, kdykoli se změní hodnota výstupního registru DAC, a tento proces vzorkování lze spustit několika způsoby.

DAC provádí inverzní operaci jako ADC, zatímco ADC převádí analogové napětí na digitální data, DAC převádí digitální čísla na analogové napětí na výstupním pinu. Ve většině STM32 je modul DAC 12bitový digitálně-analogový převodník s napětovým výstupem. DAC lze konfigurovat v 8bitovém nebo 12bitovém režimu a lze jej používat ve spojení s řadičem DMA. Ve 12bitovém režimu mohou být data zarovnána doleva nebo doprava. DAC má až dva výstupní kanály, každý s vlastním převodníkem. V režimu dvou kanálů DAC lze převody provádět nezávisle nebo současně, když jsou oba kanály seskupeny pro operace synchronní aktualizace.

Samotný výběr čipu nejvíce ovlivní na DAC vzorkovací frekvenci. Když je na výstupu DAC povolena výstupní vyrovnávací paměť, rychlost je určena kvalitou výstupní vyrovnávací paměti. Toto číslo je uvedeno pod pojmem $T_{settling}$ nebo Update rate v produktovém listu.

Když je výstupní vyrovnávací paměť deaktivována, rychlost výstupního signálu jednoduše následuje konstantu RC, která je určena výstupní impedancí DAC $R_{DAC} = 2R_a$ a kapacitní zátěží DAC_OUT.

Například u nejběžněji používaného STM32F407 definuje impedanční výstup s vypnutou vyrovnávací pamětí na maximální hodnotě 15 k Ω . Pokud se uvažuje kapacitní zátěž 10 pF, pro získání ± 1 LSB konečné hodnoty použijeme obecný vzorec 3.1, kdy po vyjádření periody T vzorec 3.2 při této konfiguraci dostáváme, že nemůže být doba převodu menší než 1,8 μ s (ekvivalentní frekvenci 555 kHz). Tato analýza však nezahrnuje žádný vliv spínací rychlosti samotného DAC a jeho přechodových jevů. Při použití vysoké rychlosti však nelze tyto faktory ignorovat. [12]

$$1 - \frac{1}{2^N} = 1 - e^{-\frac{T}{RC}} \quad (3.1)$$

$$T = CR \cdot N \cdot \ln(2) = 0.693 \cdot RC \cdot N = 1,8 \mu s \quad (3.2)$$

Výstupní data DAC STM32 je třeba zapsat do přídržného registru DAC (DHR),

poté se data přesunou do výstupního registru DAC (DOR) pro konverzi. Obecně jsou data uložena v paměti RAM a CPU má na starosti přenos dat z RAM do DAC.

Při použití DMA se celkový výkon systému zvyšuje uvolněním jádra. Důvodem je to, že data jsou přesouvána z paměti do DAC pomocí DMA, aniž by bylo potřeba jakékoli akce ze strany CPU. Tím jsou zdroje CPU volné pro další operace, což je velice užitečné při generování náhodných poruch, kdy DAC pomocí DMA generuje signál a CPU na pozadí provádí instrukce, kterými do signálu přidává náhodné poruchy, jako je například nahodný peak. Spouštění převodu DAC může být provedeno softwarem, externími spouštěči nebo časovači. Pro případy vysokorychlostního převodu se doporučuje použít spouštěč časovače v kombinaci s přenosem dat prováděným DMA. [12]

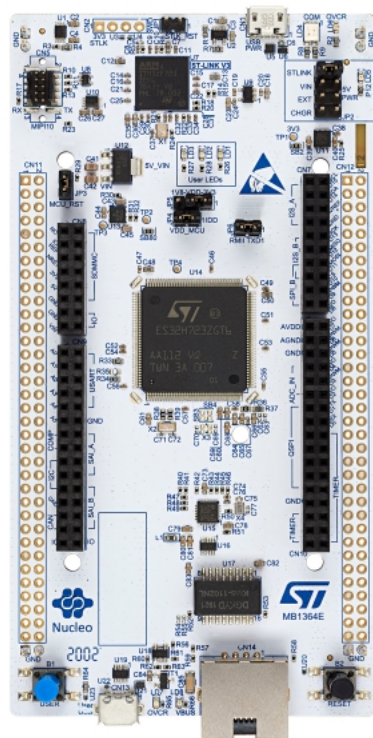
Rychlost přenosu z paměti do DAC je omezena několika faktory, mezi které patří:

- Hodinový cyklus (DAC hodiny)
- Cyklus přenosu DMA z paměti do DAC
- Samotný spoušťový mechanismus

Product	Max bus speed	DAC max sampling rate
STM32F0 Series	48 MHz	4.8 Msps
STM32F100xx	24 MHz	2.4 Msps
STM32F101xx STM32F103xx STM32F105xx STM32F107xx	36 MHz	4.5 Msps
STM32F2 Series	30 MHz	7.5 Msps
STM32F3 Series	36 MHz	4.5 Msps
STM32F40x STM32F41x	42 MHz	10.5 Msps
STM32F42x	45 MHz	11.25 Msps
STM32F7 Series	54 MHz	13.5 Msps
STM32G0 Series	64 MHz	8.0 Msps
STM32G4 Series	150 MHz	16.67 Msps 30 Msps (DMA double data mode)
STM32H7 Series	100 MHz	18.18 Msps
STM32L0 Series	32 MHz	4.0 Msps
STM32L1 Series	32 MHz	3.2 Msps
STM32L4 Series	80 MHz	10 Msps
STM32L4+ Series	120 MHz	12 Msps
STM32L5 Series	110 MHz	11 Msps

Obrázek 3.2: Porovnání frekvencí mikrokontrolérů STM[13]

Z tabulky maximálních vzorkovacích frekvencí obr.3.2, kterých lze dosáhnout na periferních zařízeních DAC různých zařízení STM32 je vidět, že největším vzorkovacích frekvencí dosahuje řada STM32H7, a proto byl vybrán čip z této řady. Při porovnání parametrů kontroléru této řady byl vybrán mikrokontrolér STM32H723. Při porovnání s mikrokontroléry ostatními této řady má jedno jádro, avšak na něm potřebnou rychlost CPU $f_{CPU} = 550\text{MHz}$ a možnost testování na testovací desce obr. 3.3 NUCLEO-H723ZG.



Obrázek 3.3: Vývojová deska NUCLEO-H723ZG[14]

4 Rešerše funkcí

V rámci práce byla provedena rešerše, kdy byly vybrány tři základní typy poruch, které jsou zajímavé jak z pohledu měření, tak i praktického využití. Se všemi se lze běžně potkat a jejich znalost je základním stavebním kamenem v měřící technice. Dále byly zvoleny tři druhy komunikací. S vývojem osciloskopů se diagnostika komunikací stává standardní funkcí, a právě praktická a názorná ukázka přenosů bitových informací popsaných níže může urychlit jejich znalost a vývoj hardwaru, který je používá.

4.1 Signál se šumem

Šum je častým problémem při návrhu elektronických obvodů i při získávání signálů pomocí osciloskopu. Šum pochází z různých zdrojů, ať už interních nebo externích, kdy může způsobit rozmazání nebo zcela ztracení hledaného signálu.

Signální šum ve svém nejzákladnějším smyslu je jakékoli nežádoucí rušení, které degraduje komunikační signál. Šum signálu může rušit analogové i digitální signály. Množství šumu potřebného k ovlivnění digitálního signálu je mnohem vyšší. Je to proto, že digitální signály komunikují pomocí sady diskretních elektrických impulsů pro přenos digitálních „bitů“. Tyto elektrické impulsy by vyžadovaly velké množství šumu, aby byly vzájemně zaměněny.

Analogové signály naopak představují nekonečný rozsah možných hodnot v rámci stanoveného rozsahu, jako je například 4 až 20 mA nebo 0 až 10 V. V tomto případě jakékoli nežádoucí napěťové nebo proudové špičky způsobí kolísání komunikované zprávy. Nepatrné variace podél analogových signálů, v řádu milivoltů nebo mik-

roampérů, obvykle nevedou k významnému (nebo dokonce znatelnému) rozporu. Vysoké úrovně elektrického šumu však mohou způsobit velké odchylky, a proto vést k podstatným nesrovnalostem, které zcela znemožňují komunikaci mezi zařízeními. [15]

4.1.1 Příčina šumu

Šum se může objevit kdekoli v systému a na jakémkoli fyzickém místě, kde jsou vodiče vystaveny vlivům prostředí. Může to být důsledkem různých faktorů. V některých případech je odstranění šumu signálu nemožné a musí se s ním na straně zpracování signálu počítat, nicméně existují některé běžné příčiny, které jsou způsobeny špatnými postupy. Tyto běžné příčiny jsou zodpovědné za převážnou většinu signálového šumu rušícího síť řízení procesů. [16]

Uzemněné smyčky a nesprávné uzemnění

Zemnicí smyčky přidávají proud do signálové smyčky prostřednictvím rozdílu napětí mezi dvěma uzemňovacími místy v systému s více uzemněními. Toto a další problémy s uzemněním mohou vést k přílivu šumu signálu do jinak funkční sítě.

Špatné postupy zapojení

Špatně zapojené sítě, jako jsou ty, které nevyužívají stíněnou kroucenou dvojlinku na vedení, jsou náchylnější k okolnímu elektrickému šumu.

Špatně navržený obvod výrobku

Špatně navržené elektronické obvody v zařízeních, které neposkytují dostatečné stínění proti vnitřním a vnějším zdrojům šumu, budou mít také větší pravděpodobnost problémů se signálem.

Blížkost jiných elektrických zařízení

Zařízení nebo vodiče umístěné v těsné blízkosti elektrických zařízení, které generují silná magnetická pole, jako jsou generátory, motory nebo elektrické vedení, mohou zachycovat část tohoto rušení, které může přispívat ke kolísání komunikačních signálů.

Dlouhé vodiče zachycující radio-frekvence

Dlouhé segmenty drátu fungují jako antény, zachycují rádiové vlny a převádějí je na elektrické signály, čímž přispívají k dodatečnému šumu v systému.

4.1.2 Redukce šumu v měření s osciloskopem

Všechny moderní digitální osciloskopy, včetně základních osciloskopů, poskytují funkce, které pomohou vypořádat se s šumem na signálu. Mezi nejběžnější patří následující tři.

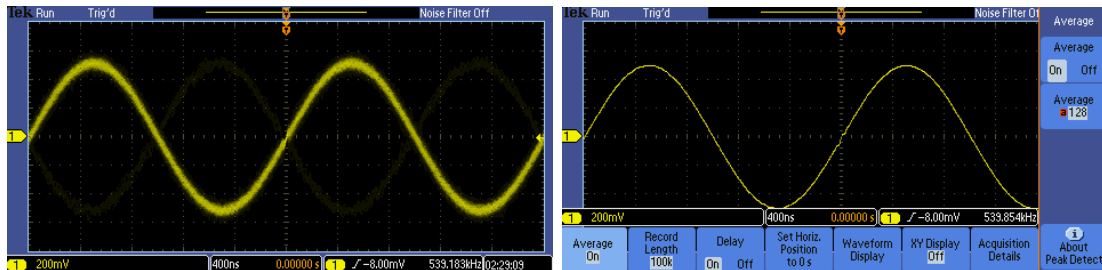
Filtrace pomocí omezení šířky pásma

Filtrace pomocí omezení šířky pásma snižuje šířku pásma osciloskopu na zvolenou frekvenci. To znamená, že frekvence vyšší než zvolená úroveň budou zeslabeny nebo zcela odstraněny. Použití tohoto filtru je jedním z nejjednodušších způsobů, jak snížit šum v signálu. Osciloskopy obvykle nabízejí omezenou sadu nastavení limitu šířky pásma. Nejběžnější výběr je 20 MHz.[17]

Režim průměrování

Šum je obvykle náhodný od bodu k bodu na stejném místě v rámci více period, někdy nahoru a někdy dolů. Proto při zprůměrování dostatečného množství periodických průběhů se šum zruší. Některé Osciloskopy mají funkci průměrného režimu snímání. Tento režim využívá data ze dvou nebo více period, která jsou zprůměrována s odpovídajícími datovými body, bod po bodu. Podmínkou pro funkčnost filtru je periodický signál.

Následující obrázek 4.1 zobrazuje na osciloskopu od firmy Tektronix, jak tato funkce funguje. Generovaný zašumělý sinusový průběh zobrazovaný v osciloskopu, bez jakéhokoli filtru se zobrazil jako nepřesný průběh. Následně byl použitý průměrovací filtr, díky kterému bylo možno získat čistý průběh ze stejného signálu.[17]



(a) Signál bez použití průměrování[17]

(b) Signál filtrovaný průměrováním[17]

Obrázek 4.1: Použití průměrovacího filtru

Matematický filtr

Některé osciloskopy nabízejí filtry pro následné zpracování, které odstraňují určité frekvence šumu ze signálu. To umožňuje plnou kontrolu nad tím, které frekvence šumu odfiltrovat. Tuto funkci nabízejí například osciloskopy od firmy Tektronix, kde se tato funkce nazývá FilterVu. Funkce pracuje jako proměnná dolní propust a umožňuje použít frekvenci dolní propusti na zobrazenou akvizici.[17]

4.1.3 Generování šumu

Pro generování šumu lze použít několik obvodů nebo lze šum generovat digitálně. Například teplotně řízené odpory, teplotně omezené vakuové diody, zenerovy diody a plynové výbojky. Generátory šumu obvykle spoléhají na základní zdroje šumu, které vytváří tepelný nebo výstřelový šum.

- **Generátor tepelného šumu**

Základním standardem může být tepelný šum. Rezistor je při určité teplotě definovaný tepelným šumem. Výsledný výstupní výkon je však nízký. Pro rezistor 1 k Ω při pokojové teplotě a šířce pásma 10 kHz je RMS šumové napětí 400 nV.

- **Generátor výstřelového šumu**

Pokud elektrony proudí přes bariéru, pak mají diskrétní časy průchodu. Tyto diskrétní průrazy vykazují výstřelový šum. Výstupní úroveň šumu generátoru výstřelového šumu se snadno nastavuje proudem elektronů skrze bariéru. Obvykle se používá bariéra v diodě nebo tranzistoru.

Digitálně generovaný šum

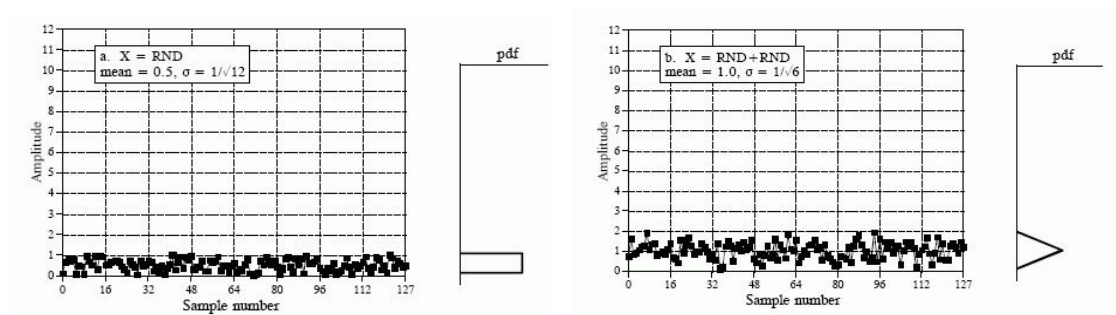
Srdcem generování digitálního šumu je generátor náhodných čísel. Většina programovacích jazyků to má jako standardní funkci. Příkaz BASIC: $X = \text{RND}$ načte proměnnou X s novým náhodným číslem pokaždé, když je funkce vyvolána. Každé náhodné číslo má hodnotu mezi nulou a jedničkou. Pravděpodobnost výskytu těchto dvou hodnot je stejná, jako kdekoli mezi těmito dvěma extrémy. Obrázek 4.2(a) ukazuje signál vytvořený zobrazením 128 vzorků z tohoto typu generátoru náhodných čísel. Průměr základního procesu, který generoval tento signál je 0,5, směrodatná odchylka $1/\sqrt{12} = 0,29$ a rozdělení je rovnoměrné mezi nulou a jedničkou.

Algoritmy je třeba testovat pomocí stejného druhu dat, se kterými se setkájí ve skutečném prostředí. To vytváří potřebu generovat digitální šum pomocí Gaussova pdf. Existují dva způsoby generování takových signálů pomocí generátoru náhodných čísel. Obr. 4.2(a) znázorňuje první metodu. Obrázek ukazuje signál získaný sečtením dvou náhodných čísel pro vytvoření každého vzorku, tj. $X = \text{RND} + \text{RND}$. Protože každé z náhodných čísel může ležet od nuly do jedné, součet musí být od nuly do dvou. Průměr je nyní jedna a směrodatná odchylka je $1/\sqrt{6}$ (když se sčítají nezávislé náhodné signály, sčítají se také rozptyly). Jak je znázorněno, pdf se změnilo z rovnoměrného rozložení na trojúhelníkové. To znamená, že signál stráví více času kolem hodnoty jedna a méně času stráveného blízko nuly nebo dvou.

Obr. 4.2(b) posouvá tuto myšlenku o krok dále přidáním dvanácti náhodných čísel k vytvoření každého vzorku. Průměr je nyní šest a směrodatná odchylka je rovna jedné. Co je ale nejdůležitější, PDF se prakticky stalo Gaussovským. Tento postup lze použít k vytvoření normálně rozloženého šumového signálu s libovolným průměrem a směrodatnou odchylkou. Pro každý vzorek v signálu:

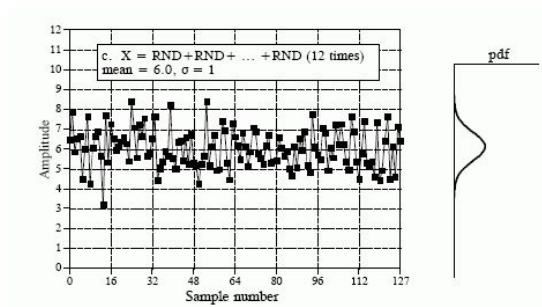
1. sečíst dvanáct náhodných čísel
2. odečíst šest, aby se průměr rovnal nule
3. vynásobit požadovanou směrodatnou odchylkou
4. přidat požadovaný průměr

Matematický základ pro tento algoritmus je obsažen v Centrální limitní větě, jednom z nejdůležitějších konceptů pravděpodobnosti. Ve své nejjednodušší podobě Centrální limitní teorém říká, že součet náhodných čísel obr.4.2(c) se stává normálně rozděleným, když se více a více náhodných čísel sčítá. Centrální limitní věta nevyžaduje, aby jednotlivá náhodná čísla pocházela z nějaké konkrétní distribuce, nebo dokonce, aby náhodná čísla pocházela ze stejné distribuce. Centrální limitní teorém poskytuje důvod, proč jsou normálně distribuované signály v přírodě tak široce zastoupené. Kdykoli na sebe působí mnoho různých náhodných sil, výsledné PDF se stane Gaussovým.



(a) Jedno náhodné číslo[18]

(b) Součet dvou náhodných čísel[18]



(c) Součet dvanácti náhodných čísel[18]

Obrázek 4.2: Grafy generátorů náhodných čísel

Ve druhém způsobu generování normálního rozdělení náhodných čísel se generátor náhodných čísel vyvolá dvakrát, aby se získaly R_1 a R_2 . Normálně distribuované náhodné číslo X pak lze nalézt:

$$X = (-2\log R_1)^{\frac{1}{2}} \cos(2\pi R_2) \quad (4.1)$$

Stejně jako dříve může tento přístup generovat normálně distribuované náhodné signály s libovolným průměrem a standardní odchylkou. Každé číslo vygenerované touto rovnicí lze vynásobit směrodatnou odchylkou a přidat požadovaný průměr.

Generátory náhodných čísel fungují tak, že začínají seedem, číslem mezi nulou a jedničkou. Když je vyvolán generátor náhodných čísel, seed prochází pevným algoritmem, jehož výsledkem je nové číslo mezi nulou a jedničkou. Toto nové číslo je hlášeno jako náhodné číslo a je pak interně uloženo, aby bylo použito jako zdroj při příštím volání generátoru náhodných čísel. Algoritmus, který transformuje seed na nové náhodné číslo.

Tímto způsobem lze generovat souvislou posloupnost náhodných čísel, všechna začínající od stejného seedu. To umožňuje, aby byl program spuštěn vícekrát za použití přesně stejných sekvencí náhodných čísel. Pokud chcete, aby se posloupnost náhodných čísel změnila, většina jazyků má ustanovení pro přenastavení generátoru náhodných čísel, což vám umožní vybrat číslo, které se použije jako první. Běžnou technikou je použití času jako seedu, čímž se poskytuje nová sekvence pokaždé, když je program spuštěn.

Z čistě matematického pohledu nemohou být čísla generovaná tímto způsobem absolutně náhodná, protože každé číslo je plně určeno předchozím číslem. K popisu této situace se často používá termín pseudonáhodný. Sekvence jsou však generované generátory náhodných čísel statisticky náhodné v mimořádně vysoké míře. Je velmi nepravděpodobné, že nastane situace, kdy náhodnost nebude adekvátní.[18]

Analogově generovaný šum

Analogově generovaný šum má výhodu oproti digitálně generovanému šumu, v jeho rychlosti generování a úplné náhodnosti jak mezi vzorky tak mezi periodami. Gene-

rování nosného signálu je lepší pomocí DAC převodníku, z důvodu možnosti výběru různých druhů signálu o snadném nastavení periody. Čistý signál na výstupu se pak spojí s šumovým signálem analogového generátoru a společně pošlou na výstup přípravku.

Druhy analogových generátorů

- **Šumový generátor se Zenerovou diodou**

Tradiční generátory bílého šumu jsou založeny na lavinovém šumu generovaném jevem Zenerova průrazu. Vznikají, když je PN přechod provozován v režimu zpětného průrazu. Lavinový šum je velmi podobný vystřelovému šumu, ale mnohem intenzivnější a má ploché frekvenční spektrum (bílý šum). Velikost šumu je obtížné předvídat kvůli jeho závislosti na materiálech.

Typický generátorový obvod se skládá ze tří částí, tj. zdroje šumu, zesilovače a přenosové části. Zdrojem šumu je lavinová dioda obr.4.3, která se zdá být dobrým kandidátem pro generování širokopásmového signálu. Lavinový průraz je jev, který se může objevit jak u izolačních, tak u polovodičových materiálů. Je to forma znásobení elektrického proudu, která může umožnit protékání velmi velkých proudů v materiálech, které jsou jinak dobrými izolanty. Protože počet elektronů v tomto jevu je Poissonova náhodná veličina, generovaný proud je pseudošumový signál. [19]



Obrázek 4.3: Zenerova dioda

- **Šumový generátor se závěrně polarizovaným emitorovým přechodem bipolárního tranzistoru**

Bipolární tranzistor obr.4.4 se závěrně polarizovaným PN přechodem má podobné vlastnosti jako Zenerova dioda. Z důvodu menších parazitních kapacit báze-emitor má však šumový generátor větší frekvenční rozsah.



Obrázek 4.4: Bipolární tranzistor

- **Šumový generátor s vakuovou diodou**

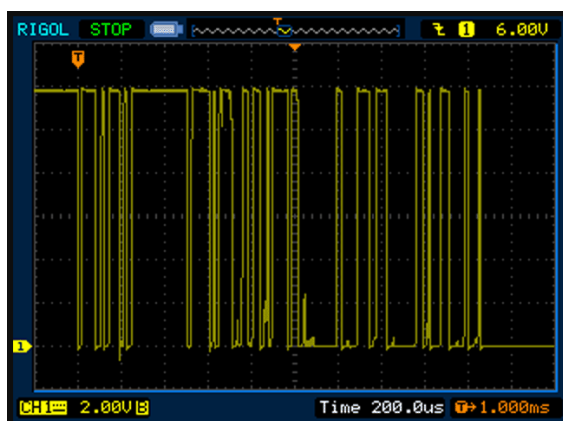
Po nějakou dobu byly (vakuové) šumové diody běžným zdrojem (bílého) šumu. Generovaly širokopásmový šum pro testování rádiových a televizních obvodů nejčastěji pomocí vakuové diody 5722 obr.4.5. Generování šumu v šumových diodách je založeno na náhodném kolísání stejnosměrného proudu (může být také interpretováno jako efekt kvantovaného transportu náboje) v saturované emisní vakuové diodě, jde tedy v podstatě o výstřelový šum. I když věci ve skutečnosti nejsou tak jednoduché, bohužel to není jediný druh šumu generovaný ve vakuové trubici, což omezuje použitelnost generátoru šumu na spodním konci frekvenčního spektra. Nejčastější problém u elektronek nastává s životností (zejména životnost žhavení, resp. katody – pokud je provozována v nasyceném stavu a odebírá nejvyšší možný katodový proud), je tím poměrně omezená a značně závisí na napětí žhavení (které následně řídí amplitudu generovaného šumu).[20]



Obrázek 4.5: Vakuová dioda 5722 od výrobce Sylvania[20]

4.2 Zákmit na hraně

Když se stiskne tlačítko, stiskne mikrospínač nebo přepne přepínač, spojí se dvě kovové části. Pro uživatele se může zdát, že kontakt je uskutečněn okamžitě. To není tak docela správně. Uvnitř spínače jsou pohyblivé části. Když stisknete spínač, nejprve se spínací kontakt dotkne druhé kovové části, ale během krátkého zlomku mikrosekundy se znova rozpojí. Pak naváže kontakt o něco déle a pak zase o něco déle. Nakonec je spínač zcela uzavřen. Přepínač přeskakuje mezi stavem sepnuto a rozepnuto. "Když je spínač zavřený, dva kontakty se ve skutečnosti oddělí a znovu spojí, obvykle 1 až 100krát za dobu asi 1 ms." ("Umění elektroniky", Horowitz & Hill, Druhé vydání, str. 506. [21]) Hardware obvykle pracuje rychleji než průběh zákmitů, což má za následek, že si hardware myslí, že je stisknutý spínač několikrát. Hardware je často integrovaný obvod. Následující snímky obrazovky obr.4.6 ilustrují typické odskoky přepínače bez jakékoliv ošetření zákmitu. [22]



Obrázek 4.6: Odskoky přepínače[23]

4.2.1 Ošetření zákmitu na hraně

V 60. a 70. letech 20. století bylo ošetření proti zákmitu na hraně implementováno pomocí různých hardwarových technik, od jednoduchých zpožďovacích obvodů odpor-kondenzátor používaných se spínači SPST až po sofistikovanější funkce latch set/reset.

V poslední době, a protože mnoho systémů obsahuje mikroprocesorovou jednotku nebo mikrokontrolérovou jednotku, stalo se běžným používáním softwarových technik k odražení signálu přicházejícího z jakéhokoli přepínače. Softwarové ošetření zákmitu na hraně však není vždy tím nejlepším přístupem. Existují některé aplikace, které zahrnují malé, nízkovýkonné, paměťově omezené procesory s omezeným kódovým prostorem a tak nevhodné pro implementaci softwarových protizákmitových funkcí. V těchto případech může být lepším řešením hardwarová implementace. [24]

Z důvodu použití hardwarových tlačítek na samotném vytvářeném zařízení bylo nutné vytvořit opatření, aby se tento jev negativně neovlivňoval chod samotného zařízení. Z důvodu dostatečného výpočetního výkonu mikrokontroleru bylo rozhodnuto o softwarovém ošetření. Tento softwarového ošetření zákmitu na hraně kódu uvedu jako příklad možného řešení problematiky.

Příklad softwarového ošetření zákmitu na hraně

Ošetření využívá časového pásma, při kterém se po zachycení hrany stává stav na tlačítku po určitou dobu zafixovaný a případný zákmit jej neovlivní. Stav na tlačítku je testován pomocí funkce "check()". Při spuštění této funkce v programu za účelem ověření, zda je některé tlačítko na zařízení zmáčknuté, se zároveň vloží jako vstupní parametr funkce port s pinem. Tímto vložáním se ve funkci definuje, stav kterého tlačítka se má kontrolovat. Nejdříve ve funkci dojde k načtení aktuálního času mikrokontroléru do paměti a následně provede podmínka, zda od minulého zmáčknutí uplynulo 50ms. Pokud ano, v následném kroku dojde k samotnému načtení stavu na tlačítku. Je-li tento stav "sepnuto" a minulý stav byl "otevřeno", dojde k negaci ve vnitřním stavu paměťového bitu tlačítka a uložení času sepnutí do paměti. Následuje opatření, aby při dlouhém držení tlačítka delším jak 50ms bylo zmáčknutí vyhodnoceno pouze jako jedno. K tomu se využívá jednoduchý princip, kdy se čeká na doběžnou hranu a teprve po ní dojde k vyhodnocení tlačítka jako TRUE neboli "sepnuto".

```
1 class Button {
2 public:
3     Button(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin, uint16_t casZpozdzeni =
4         50) :
5         _GPIOx(GPIOx), _pin(GPIO_Pin), _casZpozdzeni(casZpozdzeni) {
6         stavtlacitka = HAL_GPIO_ReadPin(_GPIOx, _pin) == GPIO_PIN_SET;
7         stavtlacitka_ts = HAL_GetTick();
8     }
9     bool check() {
10        uint32_t now = HAL_GetTick();
11        if (now - stavtlacitka_ts > _casZpozdzeni) {
12            if (stavtlacitka != (HAL_GPIO_ReadPin(_GPIOx, _pin) ==
13                GPIO_PIN_SET)) {
14                stavtlacitka = !stavtlacitka;
15                stavtlacitka_ts = now;
16            }
17        }
18    }
```

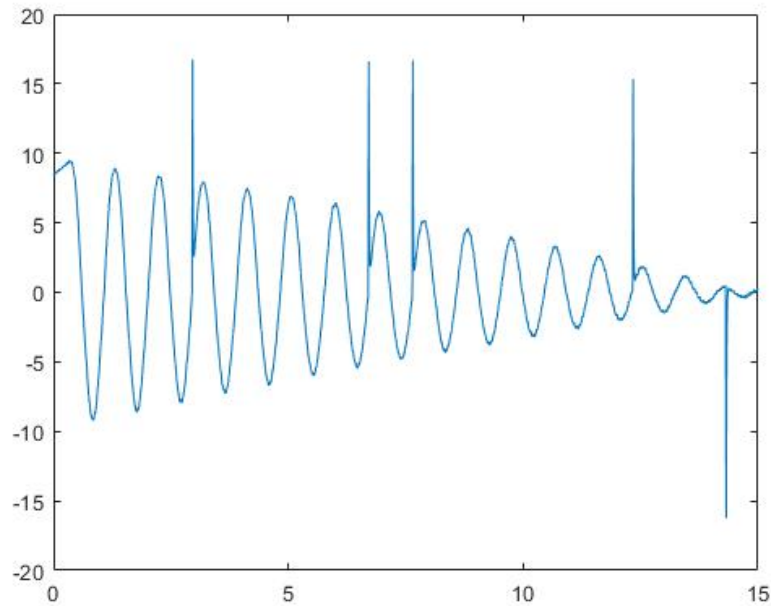
```

16     if (HAL_GPIO_ReadPin(_GPIOx, _pin) == GPIO_PIN_RESET && past == 1)
17         past = 0;
18     if (stavtlacitka == 1 && past == 0) {
19         stavtlacitka_final = 1;
20         past = 1;
21     } else
22         stavtlacitka_final = 0;
23
24     return stavtlacitka_final;
25 }
26 private:
27     uint32_t stavtlacitka_ts;
28     bool stavtlacitka, past, stavtlacitka_final;
29     GPIO_TypeDef *_GPIOx;
30     uint16_t _pin;
31     uint16_t _casZpozdeni;
32 };

```

4.3 Peaky na signálu

Dalším poměrně častým problémem jsou peaky na signálu. Peak představuje krátký interval, při kterém velikost hodnoty signálu překračují s velkým rozdílem své okolí. Tento jev lze zachytit u analogových snímačů, kdy hodnoty peaků dosahují minima či maxima rozsahu signálu. Pro znázornění problematiky zde uvést signál naměřený na 12bitovém magnetickém senzoru natočení, změřený při vypracování diplomové práce na dvojitým inverzním kyvadle [25]. Tento signál obr.4.7 díky chybě snímače obsahuje několik peaků. Pokud by se tento chybový jev dále neřešil, mohlo by dojít k velkému zkreslení, a jako v případě uvedené diplomové práce i nestabilitě regulovaného pochodu. Tento chybový jev lze odstranit hardwarovým i softwarovým řešením. Každé řešení má své klady a zápory a záleží na konkrétní problematice a možnostech implementace.



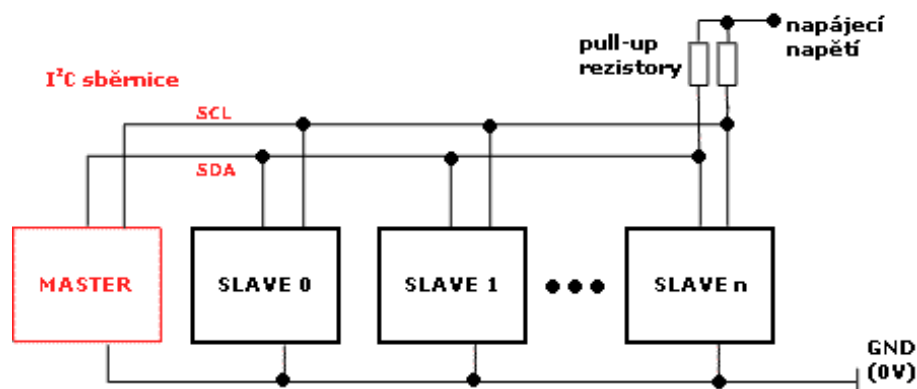
Obrázek 4.7: Peaky na reálném snímači.[25]

4.4 Komunikace I²C

Název I²C sběrnice je odvozený z původního IIC, který je zkratkou Internal-Integrated-Circuit Bus. Jak název napovídá, jedná se o sériovou sběrnici určenou pro komunikaci mezi jednotlivými integrovanými obvody. Sběrnice byla vyvinuta před 40 lety firmou Philips, která ji používala pro komunikaci na krátké vzdálenosti při nízkých rychlostech mezi integrovanými obvody a procesorem v rámci jedné desky plošného spoje. Originální komunikace byla v rámci patentu navržena na přenosovou rychlost 100 kbit/s a následně vznikly úpravy, kdy se zvyšovala její rychlost, snižovala spotřeba a upravovalo adresování. Nejnovější verze zveřejněná v roce 2007 pod názvem Ultra Fast-mode (UFm) umožňuje komunikační rychlost 5 Mbit/s s push-pull logikou oproti do té doby všemi verzemi používané logiky, kdy byly oba kanály zapojené jako otevřený kolektor. Z této komunikace byl dále v roce 2017 vytvořena komunikace I3C, která nabízí vyšší přenosové rychlosti běžnější spíše u komplikovanějších datových sběrnic jako je například Serial Peripheral Interface (SPI).

4.4.1 Fyzická vrstva

Základem této fyzické komunikace jsou dva vodiče nazývané SDA a SCL. SCL je linka hodinového taktu. Slouží k synchronizaci všech dat posílaných po této sběrnici. SDA je linka datová. Obě linky jak SCL, tak SDA propojují všechna zařízení, která je dále nutno propojit zemnicím vodičem, aby byly na stejné hladině potenciálu a také 5V pro napájení. SCL i SDA jsou při High-speed mode a modech o nižší rychlosti zapojeny jako otevřený kolektor. Z toho důvodu musí být oba vodiče implicitně v logické jedničce, což je zajištěno pull-up rezistory. Jejich odpory mají hodnotu v řádech jednotek kiloohmů. Čím je vyšší komunikační frekvence, tím musí být nižší hodnoty těchto odporů. Pro 100kHz postačuje 4k7. Maximální délka vodičů je dána nejvyšší přípustnou kapacitou v závislosti na přenosové rychlosti od 400pF po 100pF při 3,4 Mbit/s. Propojená zařízení se rozdělují na řídicí (master), které zahajují a ukončují komunikaci, generují hodinový signál a řízené (slave). Každé zařízení má specifickou 7bit adresu, jejíž kombinací je dán maximální počet zařízení na sběrnici.

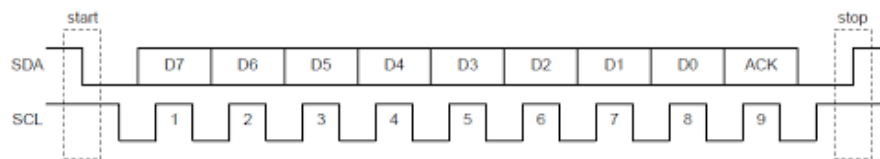


Obrázek 4.8: Princip zapojení sběrnice I²C

4.4.2 Princip přenosu

Přenos je zahájen ze stavu klid, při kterém na komunikaci neprobíhá žádný přenos dat, na obou vodičích je logická 1 způsobena pull-up rezistory, start bitem kdy se na

lince SDA změní úroveň na logickou 0 zatímco na SCL zůstává logická 1. Následuje přenos dat. Na SCL začne master generovat hodinový signál a data se přenáší na SDA po sobě jdoucích osmi bitech. Logickou úroveň na SDA lze měnit pouze při stavu logické 0 na SCL. Po 8 bitech následuje potvrzující bit Ack (acknowledge). Tento bit slouží k potvrzení správného přijetí dat. Platí pro něj stejné podmínky jako pro předchozích 8 bitů, avšak není generován vysílačem, ale přijímačem. Přenos je ukončen stop bitem, který je obdobně jako start bit vytvořen logickou 1 na lince SCL a na lince SDA se změní stav z 0 na 1.



Obrázek 4.9: Časový průběh logických úrovní na vodičích SDA a SCL

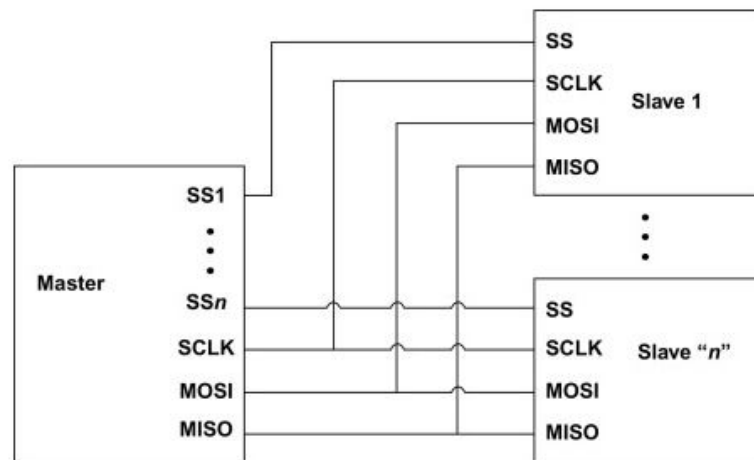
4.5 Komunikace SPI

Serial Peripheral Interface (SPI) je plně duplexní rozhraní, které ke komunikaci využívá čtyři linky. Z tohoto důvodu se SPI někdy nazývá čtyřvodičová sériová sběrnice. Čtyři vodiče jsou

- SS: Slave Select
- SCLK: Serial Clock (řízené masterem)
- MOSI: Master Out Slave In (data přecházející z master na slave)
- MISO: Master In Slave Out (data přecházející z slave na master)

Linku SS používá master k výběru podřízeného zařízení, se kterým si přeje komunikovat. Na jeden slave modul je potřeba jedna linka SS. Pokud je tedy n slave připojeno k masteru v konfiguraci sběrnice SPI, je vyžadováno n SS pinů na masteru. Linky SCLK, MOSI a MISO jsou sdíleny všemi podřízenými zařízeními. Pro

konfiguraci sběrnice, kde je n slave připojeno k masteru, bude potřeba $(3 + n)$ pinů na masteru. [26]



Obrázek 4.10: Komunikace SPI[26]

Princip přenosu

Každý přenos začíná tím, že master vybere slave pomocí SS linky. Master pak pokračuje v provozu SCLK na frekvenci menší nebo rovné maximální frekvenci podporované podřízenou jednotkou (typické rychlosti jsou v rozsahu MHz). V každém hodinovém cyklu posílá master 1 bit do slave a slave posílá 1 bit do masteru. V případě, že master nebo slave nemají žádná data k odeslání, mohou poslat 0 b.

V tomto protokolu není žádná možnost řízení toku. Master řídí hodiny, takže je může potenciálně použít k implementaci nějaké formy řízení toku, ale to se dělá jen zřídka. Slave nemá žádné prostředky k implementaci/vynucení řízení toku a očekává se, že zpracuje všechna data, která obdrží.

Pro protokol SPI není definována žádná standardní sada rychlostí. Typické rychlosti sběrnice jsou v rozsahu 50 MHz. Protože protokol nepřidává žádnou režii, jako je adresování ani řízení toku, a tak propustnost, které lze dosáhnout pomocí SPI odráží frekvenci hodin. U linky 50 MHz SPI je propustnost 50 Mbps. Komunikace je plně duplexní. Efektivní propustnost tedy ve zmíněném případě může dosahovat až 100 Mbps (pokud si master i slave dokážou posílat smysluplná data).

4.5.1 Porovnání s I2C

Klíčovou výhodou protokolu SPI je, že podporuje velmi vysoké rychlosti a tím i propustnost. Typické rychlosti sběrnice SPI jsou v rozsahu 50 MHz a s absencí jakékoli režie protokolu, jako v případě I2C, se to přesně promítá do bitové rychlosti. Díky tomu je sběrnice SPI nejvhodnější pro snímače, které produkují velké množství dat, jako jsou například snímače otisků prstů.

V protokolu SPI je standardně povolena plně duplexní komunikace. Pokud nastane situace, kdy si master i slave potřebují navzájem přenášet stejné objemy dat, lze tuto funkci využít.

Protokol SPI je ve své složitosti srovnatelný s protokolem I2C – oba jsou poměrně jednoduché. V SPI musí být vzorkování datové linky a změna datových hodnot synchronizována s náběžnou a sestupnou hranou hodinové linky. Běžné je také využití bit-banging (softwarového řízení komunikace) protokolu SPI pomocí GPIO pinů, jako je tomu u I2C. Díky absenci pull-up rezistorů, které jsou přítomny v případě I2C, bude spotřeba energie rozhraní SPI sběrnice nepatrně nižší než u rozhraní I2C.

Protože protokol pro SPI neobsahuje žádný mechanismus řízení toku, je proto nutné, aby software zpracoval přijatá data rychleji, než dorazí další. Tento problém je zmírněn přítomností vyrovnávací paměti FIFO (First In First Out) na přijímací straně. Protokol SPI neobsahuje žádné kontroly o detekci chyb a opravné kódy. V případě potřeby musí být detekce/oprava chyb implementována v dalším softwaru.

V SPI není žádné vnitropásmové adresování (na rozdíl od I2C). Každý SPI slave, který je připojen k masteru, bude potřebovat vyhrazený SS pin na masteru. To znamená, že počet pinů požadovaných na masteru se bude zvyšovat se zvyšujícím se počtem připojených slave. Pro SPI master je typické ovládat dva až tři slave a ne více. Neexistuje také žádný formální standard pro protokol SPI, to znamená, že neexistuje žádný hodnocení pro ověřování shody daného produktu s protokolem.

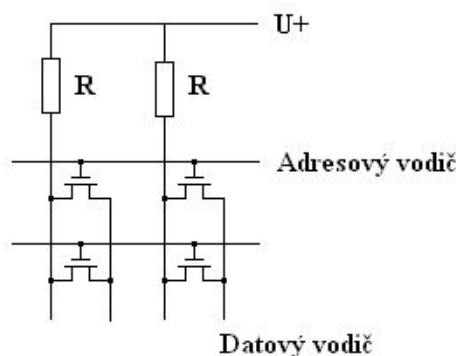
4.5.2 Paměť EEPROM

EEPROM (Elektricky vymazatelná programovatelná paměť pouze pro čtení) Přepisovatelný úložný čip obr. 4.11, který uchovává svůj obsah bez nutnosti stálého napájení. Paměti EEPROM jsou adresovatelné po bajtech, ale před přepsáním je nutné je vymazat. Ve flash paměti, která se vyvinula z EEPROM a je architektonicky téměř identická, avšak je mazatelná po celých blocích bajtů. Kromě toho se paměti EEPROM obvykle používají na deskách plošných spojů k ukládání malého množství instrukcí a dat, zatímco moduly flash paměti uchovávají gigabajty dat. Paměti EEPROM ukládají data o adresaci v páskových kazetách LTO a také se používají v jednotkách SSD založených na flash paměti, ne však jako úložiště dat, ale pro uložení kódu pro spuštění.



Obrázek 4.11: Paměť EEPROM

Bitové buňky v EEPROM a flash paměti jsou tranzistory založené na CMOS. Při zápisu dat se přivede na příslušný adresový vodič záporné napětí $-U$ a datový vodič buněk, do nichž se má zaznamenat se uzemní. Tranzistor se otevře a vznikne v něm náboj, který vytvoří velké prahové napětí. Při čtení se přivede na adresový vodič záporný impuls. Tranzistor s malým prahovým napětím se otevře a vede elektrický proud do datového vodiče, zatímco tranzistor s velkým prahovým napětím zůstane uzavřen. Vymazání paměti se provádí kladným napětím $+U$, které se přivede na adresové vodiče. Náboj se tím zmenší a prahové napětí poklesne, čímž je paměť vymazána. [27]



Obrázek 4.12: Realizace buňky paměti EEPROM pomocí tranzistoru[28]

Paměť EEPROM má životnost mezi 10 000 a 100 000 cyklů zápisu, což je podstatně více než u paměti EPROM (jedno "E"), které jim předcházely.

4.6 Paralelní komunikace

Paralelní komunikace je způsob odesílání několika datových signálů současně přenosovým spojem v jednom okamžiku. Skládá se z několika paralelně propojených kanálů. V paralelním režimu přenosu dochází k jednorázovému přenosu dat ze zdroje do cíle. To je možné pomocí více kanálů pro přenos dat mezi vysílačem a přijímačem.

Základní rozdíl mezi paralelním a sériovým kanálem je jeho množství různých vodičů ve fyzické podobě, které se používají pro přenos dat mezi různými zařízeními. Na rozdíl od sériového přenosu využívá paralelní přenos více než jeden vodič (a to s výjimkou zemního vodiče).

Tento způsob komunikace je drahý, protože vyžaduje další kabely a hardware. Nicméně je to považováno za rychlý způsob přenosu. Tiskárny, RAM, ISA, ATA, SCSI, PCI, Front Side Bus, atd. používají paralelní přenos dat. Klesající náklady na integrované obvody a zvyšující se poptávka spotřebitelů po vyšší rychlosti přenosu dat z nich učinily v některých případech vhodnou volbu oproti sériovým linkám.[29]

Princip přenosu

Při paralelní komunikaci se mnoho bitů (normálně 8 bitů nebo jejich násobky) přenáší současně na různých paralelních cestách (vodičích) v rámci stejného kabelu v synchronizaci s jediným hodinovým taktem. Hodiny poskytují časování pro přenos jako konstantní hodinový signál přes paralelní cesty.

Protože je současně přenášeno mnoho bitů po různých paralelních cestách, může být pořadí přijatých bitových řetězců odlišné nebo nesynchronizované v závislosti na různých faktorech, jako je vzdálenost zdroje, umístění a dostupná šířka přenosového pásma. Příkladem toho je zkreslení/přeskakování při internetových hovorech a streamování videa.

4.6.1 16x2 LCD displej

LCD obrazovka je elektronický zobrazovací modul, který využívá tekuté krystaly k vytvoření viditelného obrazu. LCD displej 16x2 obr.4.13 je základní modul běžně používaný v jednoduchých obvodech. Jak název displeje 16x2 prozrazuje, jedná se o zobrazení 2 řádků a 16 znaků na řádek. Na tomto LCD je každý znak zobrazen v matici 5x7 pixelů.



Obrázek 4.13: LCD display s integrovaným obvodem HD44780

Displej 16x2 obsahuje paralelní rozhraní, čehož se využívá pro ukázkou paralelní komunikace. Rozhraní obsahuje piny:

- Register Select pin

- Read/Write pin
- Enable pin
- datové piny od D0 do D7
- pin kontrastu displeje
- piny podsvícení LED
- piny napájení

Proces ovládání displeje tvoří především zápis bitů do datového registru pro vytvoření symbolu, který se má zobrazit, a také umístění instrukcí do registru instrukcí, například o místě kurzoru, řádku a pozici. Na internetu lze nalézt mnoho knihoven, jako je například Liquid Crystal, které práci s displejem zjednoduší, takže není třeba ovládání displeje na nízké programovací úrovni. Ovládání LCD čipem HD44780 od výrobce Hitachi lze provádět pomocí dvou režimů, 4bitový nebo 8bitový. Pro 4bitový režim je nutné při zapojení použít 7 I/O pinů nebo 8bitový režim který vyžaduje 11 pinů. Pro zobrazení textu na LCD v této práci se používá 4bitový režim.[30]

Princip činnosti LCD

LCD obrazovka se skládá ze dvou částí, skutečného displeje z tekutých krystalů a zdroje světla na zadní straně obrazovky (tzv. podsvícení). Mezi podsvícením a obrazovkou LCD je umístěn difuzér světla, aby byl zdroj světla stejnoměrný po celé obrazovce.[31]

LCD panel sám o sobě nevyzařuje světlo, a proto potřebuje podsvícení. Funguje pouze jako filtr, který blokuje světlo na základě jednotlivých pixelů. Podsvícení je vždy zapnuté a pixely na displeji se otáčejí, aby propouštěly světlo, čímž se vytvářejí barvy potřebné pro obraz. Pokud chce obrazovka zobrazit černou barvu, pixely LCD se otočí, aby se pokusily zcela zablokovat světlo. Pokud chce zobrazit bílou, propustí všechno světlo. Vzhledem k tomu, že displej je pouze filtr, černá nebude tak hluboká,

protože LCD panel vždy propustí malou část světla. V současné době se LCD často používají u televizí, v digitálních hodinkách, počítačích atd. V průmyslu obrazovky LCD nahradily CRT (Cathode Ray Tubes), protože tyto obrazovky spotřebují více energie než LCD, jsou těžší a větší. Displeje LCD jsou ve srovnání s CRT tenčí avšak ve srovnání s OLED obrazovkami má LCD vyšší spotřebu energie, protože funguje na základním principu blokování světla, díky čemuž je nutný stálý podsvit.[32]

5 Praktická implementace funkcí

Díky dostatečnému výpočetnímu výkonu se všechny funkce řeší softwarově. Umožňuje to tak vysokou kontrolu a široké nastavení generovaných funkcí bez nutnosti složitých analogových obvodů. Proto se přesunula velká část práce do programovacího prostředí, kde se odehrává veškeré generování a simulování. Práce samotná tedy nejvíce spočívá ve vytváření programu mikrokontroléru, implementaci funkcí na nastavování, generování signálu a práci s Halovými knihovnami pro komunikace a využití analogového převodníku na výstupu.

5.1 Generování šumu

Pro výukové účely zařízení dokáže generovat signál s předem nastavenou hranicí šumu. Při vyvíjení tohoto zařízení byl kladen velký důraz na spolehlivou a kontrolovatelnou funkčnost za všech podmínek. Proto se generátor šumu nezakládá na náhodnosti z vnějšího prostředí. Šum lze generovat mnoha způsoby, například jej generovat v závislosti na radiových vlnách snímaných na anténě nebo v závislosti na hluku v okolí. Tyto jevy však jsou obtížně normalizovatelné, a tak i nežádoucí. Dalším zdrojem šumu jsou elektronky, u nich je však větší energetická náročnost, z důvodu nutnosti zdroje žhavicího napětí. Nejlepšími variantami generování šumu pro toto použití tak poskytují polovodičové přechody a digitální šum. V zařízení bylo nakonec vybráno, z důvodu možnosti širšího nastavení, generovat šum digitálně.

Možnosti nastavení

- Typ šumu

- Spojitý šum
- Gaussův šum
- Funkce signálu
 - Sinusový signál
 - Pilový signál
 - Trojúhelníkový signál
 - Obdélníkový signál
- Frekvence signálu 10Hz -10kHz
- Velikost šumu 0-100%

Digitálně generovaný šum

Digitálně generovaný šum je založený na generátoru náhodných čísel. Mikrokontrolér STM32H7 má v sobě tento generátor integrovaný. Generátor je založena na spojitém analogovém šumu, který poskytuje náhodnou 32bitovou hodnotu. Je schopen generovat čtyři 32bitová náhodná čísla s minimální frekvencí 216 cyklů systémových hodin. Generátor je založen na analogovém obvodu, který složen z několika prstencových oscilátorů, na jejichž výstupech jsou odebírány vzorky a poté XORovány, aby se vytvořil seed. Seed se vkládá do digitálního post-processingového bloku, který je schopen vyrobit čtyři 32bitová náhodná čísla za cyklus výpočtu. Vzorkování analogovým seedem je taktováno nezávislým zařízením na hodinový signál tak, aby kvalita náhodného čísla byla nezávislá na frekvenci hodinového taktu hlavního čítače mikrokontroléru. Obsah bloku následného zpracování se přenesse do datový registr prostřednictvím čtyř-slovného FIFO registru. Příznak připravenosti se spustí, jakmile je FIFO plný a automaticky se resetuje. [33]

Takto vygenerované náhodné 32bitové číslo se poměrově zmenší na nastavenou hodnotu rozptylu okolo nosného signálu. DAC vytváří analogový signál s 12bitovým rozlišením. Nosný signál zabírá z důvodu rozptylu šumu nad velikost maximální

amplitudy půlku celkového pásma tedy 11bitů. Postup se odvíjí od nastavení typu šumu. Při nastavení spojitého šumu je vygenerované náhodné 32bitové číslo. Toto číslo se nejdříve rozdělí na dvě poloviny z důvodu možnosti záporného šumu vůči nosnému signálu a pak poděleno 20bitovým číslem. Tím se získá náhodná hodnota o 11bitech, tedy 100% rozsahu nosného signálu. Dále je zmenšeno na nastavený poměr šumu v procentech. Program si uloží minulou hodnotu šumu a pokud se následující změny o limitní mez, hodnota šumu je na tuto mez nastavena. Tím se docílí toho že jej bude osciloskop vidět jako spojitý signál. Následně se hodnota šumu přičte nebo odečte k nosnému signálu v závislosti na 32.bitu z původního náhodného čísla.

```
1 void calcspojsum(uint16_t pvelsum, uint16_t lim) {
2 //pvelsum – uroveň šumu v procentech
3 //lim – maximální změna v signálu
4 uint32_t nahoda;
5 uint16_t sum;
6 uint32_t summin;
7
8 if (pvelsum == 0)
9     return;
10
11 uint32_t pom = 100 / pvelsum;
12
13 for (int i = 0; i < 1000; i++) {
14     while (HAL_RNG_GenerateRandomNumber(&hrng, &nahoda) != HAL_OK) {}
15     if (nahoda > 2147483648) {
16         nahoda = nahoda - 2147483648;
17         sum = nahoda / 1048576 / pom;
18         summin = sum;
19         if ((sum - summin) >= lim)
20             sum = sum + lim;
21         if ((sum - summin) <= lim)
22             sum = sum - lim;
23         signal_do_DAC[i] = signal_do_DAC[i] / 2 + 4096 / 4 + sum;
24     } else {
```

```

25     sum = nahoda / 1048576 / pom;
26     summin = sum;
27     if ((sum - summin) >= lim)
28         sum = sum + lim;
29     if ((sum - summin) <= lim)
30         sum = sum - lim;
31     signal_do_DAC[i]=signal_do_DAC[i]/2+4096/4-sum;
32 }}}

```

Pokud je vybrán šum z Gaussova rozdělení, nejdříve se pro vytvoření šumu, o směrodatné odchylce rovné jedné, sečte dvanáctkrát vygenerované náhodné 32bitové číslo zmenšené na maximální rozsah rozlišení AD převodníku, což je 12bitů. Vznikne tak náhodná hodnota z Gaussova rozdělení se střední hodnotou o dvanácti násobku hodnoty poloviny rozsahu a rozsah roven dvanácti násobku 12bitového čísla. Následně je podělena dvanácti, čímž se docílí toho, že střední hodnota bude stále v polovině a rozsah bude odpovídat rozsahu převodníku. Získaná hodnota se následně jako v předchozím typu generování šumu rozdělí z důvodu možnosti záporného šumu vůči nosnému signálu na dvě poloviny, zmenší dle nastavené procentuální hladiny šumu a přičte nebo odečte k nosnému signálu.

```

1 void calcgaussum(uint16_t pvelsum) {
2 //pvelsum - uroven sumu v procentech
3 //lim - maximalní zmena v signálu
4     uint32_t nahoda;
5     uint16_t sum;
6     uint16_t gaussum=0;
7
8     if (pvelsum == 0)
9         return;
10
11     uint16_t pom = 100 / pvelsum;
12
13     for (int i = 0; i < 1000; i++) {
14         for (int gaus = 0; gaus < 12; gaus++) {
15             while (HAL_RNG_GenerateRandomNumber(&hrng, &nahoda) != HAL_OK)
16                 {}

```



```

16     sum = nahoda / 1048576;
17     gaussum=gaussum+sum;
18 }
19 gaussum=gaussum/12;
20 if (gaussum > 2048) {
21     gaussum = gaussum - 2048;
22     gaussum=gaussum/pom;
23     signal_do_DAC[i] = signal_do_DAC[i] / 2 + 4096 / 4 + gaussum;
24 } else {
25     gaussum = 2048 - gaussum;
26     gaussum=gaussum/pom;
27     signal_do_DAC[i] = signal_do_DAC[i] / 2 + 4096 / 4 - gaussum;
28 }
29 }
30 }

```

Takto je vygenerovaný šum na signálu o jedné periodě. Takovéto generování vzorků signálu je časově náročnější, a proto frekvence generování vzorků je nižší než frekvence změny vzorků na výstupu z DAC. Z toho důvodu nelze vzorky generovat v reálném čase a musí se vygenerovat dopředu. Pokud by se vygenerovala pouze jedna perioda zašuměného signálu a pustila se na výstup dokola, na osciloskopu by to v normálním režimu nebylo vidět, ale pokud bychom použili funkce pro jejíž otestování a naučení se s nimi je toto zařízení určené, například odstranění šumu pomocí průměrování z více period. Signál by se díky stejným hodnotám v stejných místech napříč periodami nezměnil. Proto je třeba předem do paměti vygenerovat dostatečný počet period. Funkce proto vygeneruje 70 period zašumělého signálu které uloží do paměti, ze které následně DAC s pomocí DMA vytvoří signál na výstupu.

```

1 void calcsin_velky(uint16_t ppvelsum, uint16_t pfsun, uint16_t plim) {
2
3     for (int u = 0; u < 70; u++) {
4         switch (pfsun) {
5             case 1:
6                 calcsin();

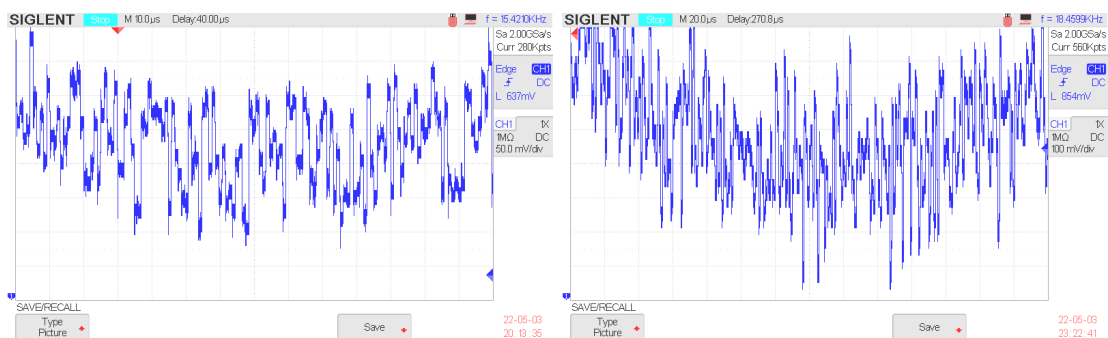
```

```

7     break;
8     case 2:
9         calcpila();
10        break;
11        case 3:
12            calctroj();
13            break;
14        case 4:
15            calcobd();
16            break;
17        default:
18            calcsin();
19            break;
20    }
21    if(sumTyp)
22        calcspojsum(ppvelsum, plim);
23    else
24        calcgaussum(ppvelsum);
25
26    for (int v = 0; v < 1000; v++) {
27        signal_do_DAC_sum[u * 1000 + v] = signal_do_DAC[v];
28    }}}

```

Při porovnání takto generovaných šumů lze na obr.5.1 vidět, že šumy odpovídají jejich teoretickým předpokladům. Spojitý šum na obr.5.1(a) neobsahuje velké množství peaků a signál se mění jen po limitovaných krocích, oproti tomu na obr.5.1(b), kde je signál Gaussova šumu. Gaussův šum se nejvíce pohybuje okolo své střední hodnoty a signál obsahuje rychlé, nelimitované peaky.

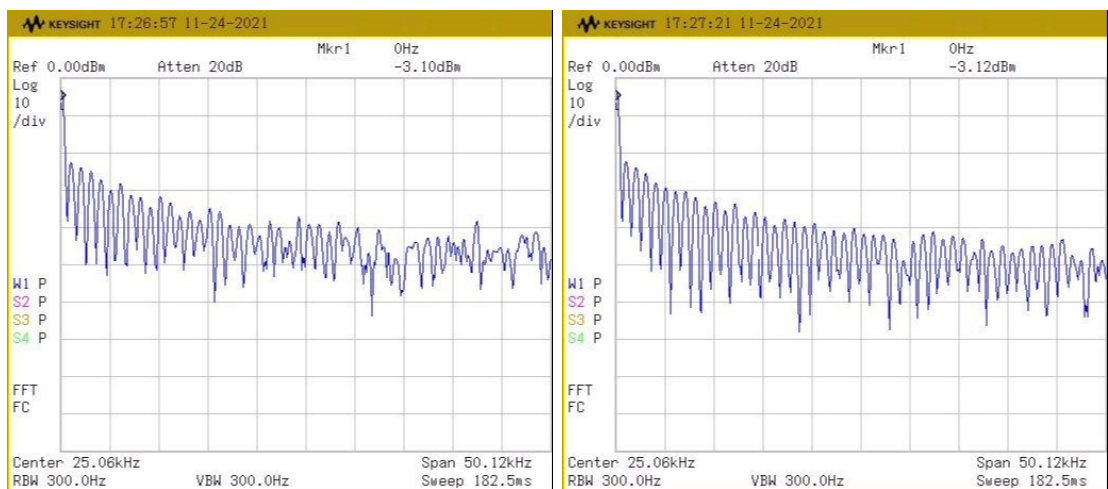


(a) Spojitý šum

(b) Gaussův šum

Obrázek 5.1: Vytvořený šum

Pro otestování kvality vytvořených šumů byly oba generované signály měřené spektrálním analyzátozem. Jak je možné vidět na obrázcích 5.2, oba signály obsahují frekvence v celém měřeném pásmu, avšak při bližším porovnání frekvencí horní části spektra lze vyhodnotit Gaussův šum, díky pravidelnějšímu uspořádání peaků, jako dokonalejší.



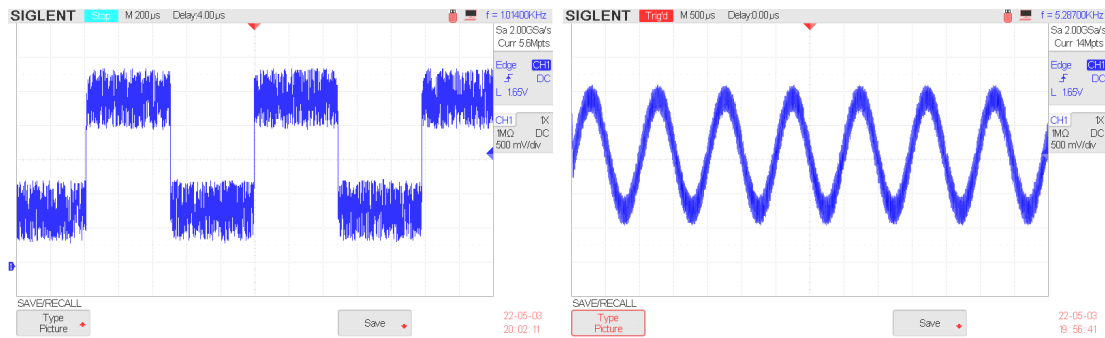
(a) Spektrum spojitého šumu

(b) Spektrum Gaussova šumu

Obrázek 5.2: Porovnání spekter šumů

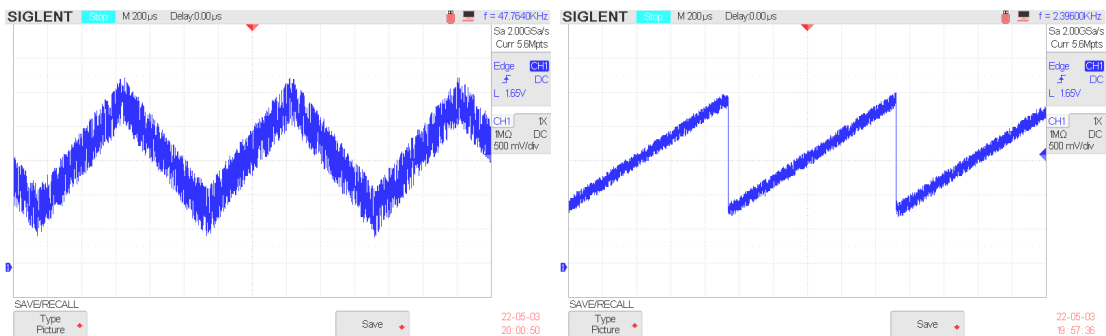
Na obrázcích obrázku 5.3 lze vidět generovaný šum při různém nastavení zařízení. Pro prezentaci možností generátoru byly vygenerovány čtyři průběhy o odlišném nastavení. První obrázek 5.3(a) zobrazuje šum na obdélníkovém signálu generovaný pomocí Gaussova šumu, nastavený na vysokou úroveň zašumění. Druhý obrázek

5.3(b) šum na sinusovém signálu generovaný pomocí také Gaussova šumu, ale o menší intenzitě než první. Třetí obrázek 5.3(c) šum na trojúhelníkovém signálu o vysoké intenzitě zašumění, už však generovaný pomocí spojitého šumu a čtvrtý obrázek 5.3(d) pilového signálu o nízké intenzitě spojitého šumu.



(a) Šum na obdélníkovém signálu

(b) Šum na sinusovém signálu

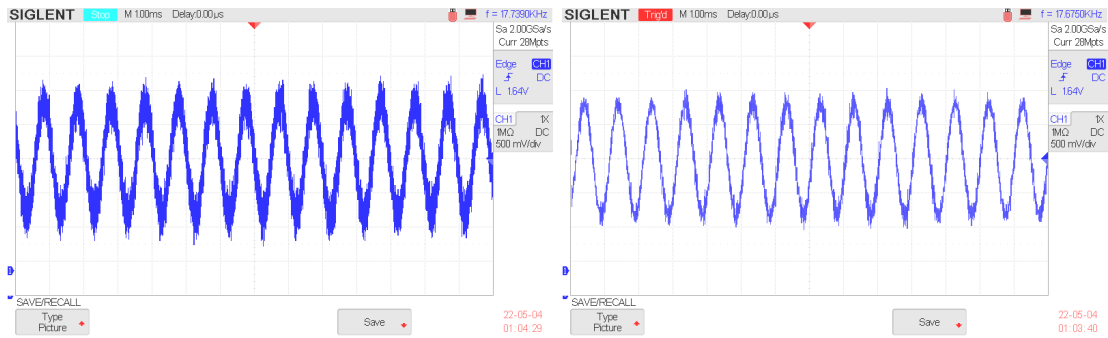


(c) Šum na trojúhelníkovém signálu

(d) Šum na pilovém signálu

Obrázek 5.3: Generované šumy

Pro otestování správné funkčnosti generování v rámci více period bylo na zařízení nastaveno generování Gaussova šumu na sinusovém signálu, osciloskop přepnut do režimu normálního zobrazení, a pak následně při stejném signálu do průměrovacího režimu. Při porovnání zachycených průběhu na obr.5.4 lze vidět snížení úrovně šumu díky průměrování.



(a) Zobrazení v normálním režimu

(b) Zobrazení v průměrovacím režimu

Obrázek 5.4: Porovnání průběhu po průměrovacím režimu

5.2 Vytváření zákmitu na hraně

Simulace zákmitu na hraně je důležitou funkcí tohoto zařízení hlavně z důvodu obtížné predikci tohoto jevu. Při vývoji jakéhokoliv zařízení se musí s tímto jevem počítat, ale je jen málo způsobů, jak otestovat, zda toto řešení skutečně funguje. Při vývoji tohoto zařízení bylo testováno mnoho spínačů či mikrospínačů, avšak zákmit nastal jen v zlomku případů. Pravděpodobnost objevení se jevu při náhodně vybraném spínači je tedy poměrně malá, ale pokud by se tento jev objevil, může dojít k nefunkčnosti celého zařízení, což může mít dále fatální následky. Ze zachycených zákmitů byl udělán rozbor a porovnání, z čehož byly vyvozeny parametry vhodné pro nastavení v simulaci.

Možnosti nastavení

- Délka zákmitu
- Počet zákmitů
- Tlumení

Funkce na softwarové vytvoření zákmitu na hraně není oproti ostatním funkcím tohoto zařízení nikterak složitá. V paměťovém poli DAC převodníku se nejdříve

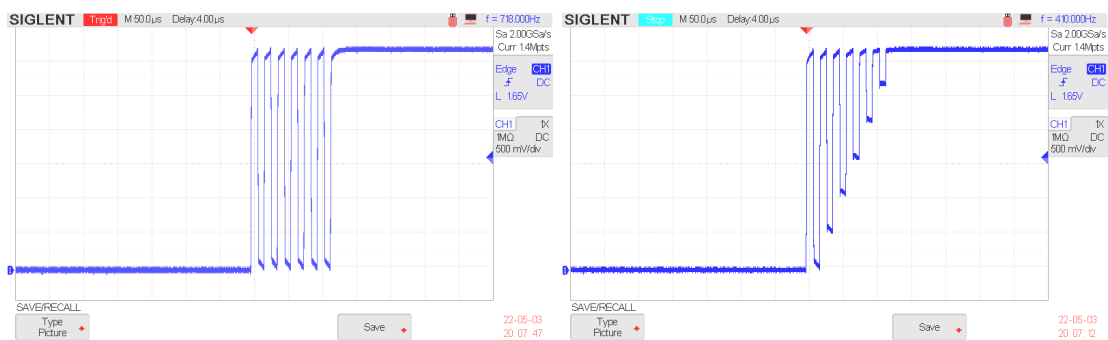
vytvoří čistý přechod mezi dvěma hladinami. Tento přechod se dále upravuje podle nastavených hodnot. Za první náběžnou hranou se po nastavené délce vytvoří hrana sestupná, a dále opět náběžná. Při nastavení více zákmitů se celý proces opakuje s posunem vždy dvou délek zákmitu. Pokud je zapnuté tlumení, signál se při zákmitech nevrací na původní hodnotu, ale na hodnotu která je odvozená od počtu zákmitů.

```

1 void calczakmit(uint16_t pDelka, uint16_t pPocet, bool pTlumeni) {
2     for (int i = 0; i < 1000; i++) {
3         if (i < 300)
4             signal_do_DAC[i] = 0;
5         else
6             signal_do_DAC[i] = 4095 ;
7     }
8     for (int i = 0; i < pPocet; i++) {
9         for (int a = 0; a < pDelka; a++) {
10
11             signal_do_DAC[i * pDelka * 2 + a+300] = 4095 ;
12         }
13         for (int a = 0; a < pDelka; a++) {
14             if (!pulsTlumeni)
15                 signal_do_DAC[i * pDelka * 2 + pDelka + a+300] =(4095*i)/pPocet;
16             else
17                 signal_do_DAC[i * pDelka * 2 + pDelka + a+300] = 0;
18         }
19     }
20 }

```

Pro ukázkou vytvořeného zákmitu bylo nastaveno šest zákmitů o krátké délce obr.5.5(a), a pak následně doplněn signál o tlumení obr.5.5(b).



(a) Peak na pilové signálu

(b) Peak na trojúhelníkovém signálu

Obrázek 5.5: Zákmit na signálu

5.3 Peak na signálu

Pro otestování funkcí na hledání peaků v signálu nebo pro možnost se naučit s těmito funkcemi pracovat dokáže zařízení vytvořit signál a na něm následně tyto peaky generovat.

Možnosti nastavení

- Funkce signálu
 - Sinusový signál
 - Pilový signál
 - Trojúhelníkový signál
 - Obdélníkový signál
- Frekvence signálu 10Hz -10kHz
- Délka peaku (délka v procentech z periody)
- Četnost peaku (šance v promilích na peak za periodu)

Generování peaku

Při generování peaku se využívá vlastnosti DMA a DAC, kdy jeho generování analogových hodnot na výstupu pracuje paralelně a nezávisle na chodu hlavního programu. Pro výpočet a generování je použita funkce z HALových knihoven. Tato funkce je automaticky spuštěna s resetováním DAC po průběhu celého pole hodnot signálu. V paměti je předem uložen signál jedné periody o tisíci vzorcích. Tato funkce se tak provede každou periodu a pomocí generátoru náhodných čísel určí, zda se bude v následující periodě generovat peak a na jakém místě bude peak umístěn.

Funkce po zavolání nejdříve zkontroluje, zda není v paměti už nějaký peak uložen. Děje se to, protože se provádí úpravy na paměti, ze které se souběžně generuje signál, a tak by její kompletní přehrání nebylo možné z důvodu rychlosti a možnosti chyb při přehrávání a načítání stejného místa paměti. Pokud se zjistí, že není v paměti uložen žádný peak, spustí se generátor náhodných čísel. Výsledné náhodné číslo o velikosti 2^{32} je vyděleno tak, aby bylo v rozsahu 1 až 1000 a porovnává se s velikostí hodnoty nastavené pro četnost peaků. Pokud je náhodné číslo menší jak četnost peaků, funkce vygeneruje peak nastavené délky. Pozice peaku v signálu je náhodná a závisí na generovaném náhodném čísle z minulé periody. Před zápisem peaku na signál se uloží hodnoty, které se nachází na pozici generovaného peaku, do pomocné paměti. Následující průběh periody tyto hodnoty vrátí na své původní místa a peak se tím odstraní.

```
1 void HAL_DAC_ConvCpltCallbackCh1(DAC_HandleTypeDef *hdac) {
2
3     if (peakStav == 1) {
4         for (int i = 0; i < peakDelka*10; ++i) {
5             signal_do_DAC[peakNahoda_old + i] = peakPamet[i];
6         }
7         peakStav = 0;
8     }
9     while (HAL_RNG_GenerateRandomNumber(&hrng, &nahoda) != HAL_OK) {}
10
11     peakNahoda_old=peakNahoda;
```

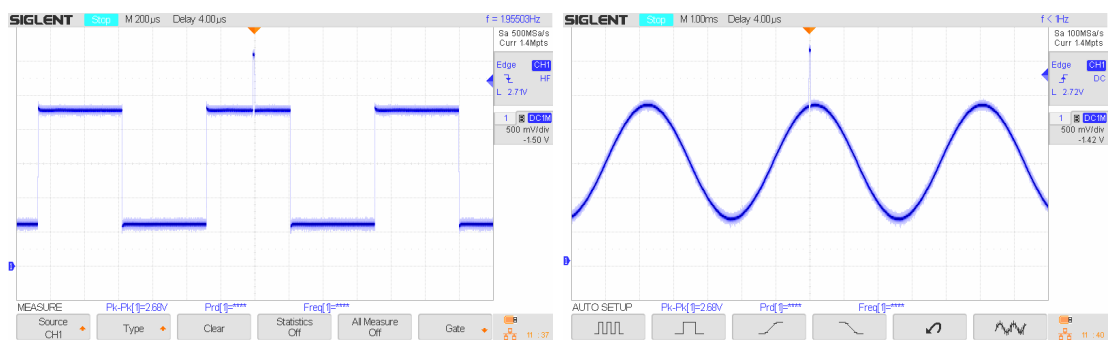


```

12 peakNahoda = nahoda / 4294967;
13
14 if (peakNahoda < peakPocet) {
15     for (int i = 0; i < peakDelka*10; ++i) {
16         peakPamet[i] = signal_do_DAC[peakNahoda_old + i];
17         signal_do_DAC[peakNahoda_old + i] = 4095;
18     }
19     peakStav =1;
20 }
21 }

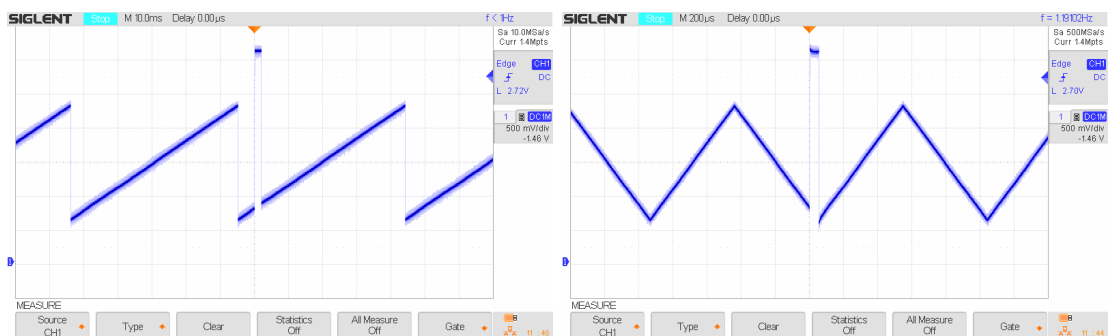
```

Na obrázcích 5.6 lze vidět různé nastavení generátoru peaku na signálu. Peak je generován na odlišných nosných signálech, nastavených rozdílných periodách i délkách peaku.



(a) Peak na obdelníkovém signálu

(b) Peak na sinusovém signálu



(c) Peak na pilové signálu

(d) Peak na trojúhelníkovém signálu

Obrázek 5.6: Peak na signálu

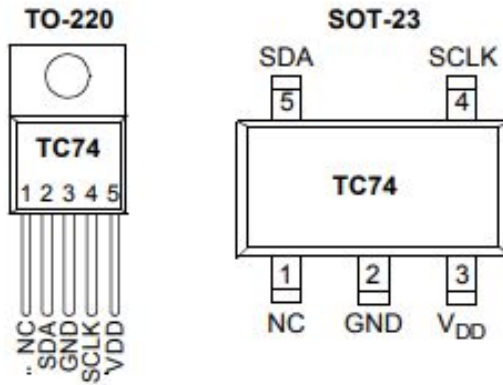
5.4 Simulace I2C

Pro simulaci komunikace byl využit digitální teplotní senzor s komunikací I2C. Tento senzor je osazen na desce plošného spoje, kde je spojen jak s mikrokontrolérem, tak s piny pro připojení sondy osciloskopu umístěnými na jejím okraji. Komunikaci řízenou mikrokontrolérem lze nastavením v menu upravit i tak, že se senzor zcela znefunkční. To se může stát při výběru komunikační rychlosti, kterou teplotní senzor nepodporuje nebo změněním adresy zařízení. Toto lze následně využít k ověření znalosti studenta, který na osciloskopu zjišťuje příčinu poruchy.

Možnosti nastavení

- Rychlost komunikace
 - 100 kbps standard mode
 - 400 kbps fast mode
 - 1 Mbps fast mode +
- Adresa slave zařízení (předem nastavená správná, lze pro výukové účely a zobrazení změnit)
- Bytová délka čtených dat

Jako digitální teplotní senzor byl využit čip TC74. Tento digitální teplotní senzor od Microchip Technology získává a převádí teplotní informace ze svého zabudovaného polovodičového senzoru s rozlišením 1°C a rychlostí měření 8 vzorků/s. Přesnost tohoto čipu je relativně nízká, ale to je k účelu zobrazení komunikace I2C je nepodstatné, a naopak velkou výhodou je velká rozšířenost, kvalitní a srozumitelně zpracovaná technická dokumentace, která usnadní výukové účely. Teplota je k dispozici jako 8bitové digitální slovo uložené v interním teplotním registru, který je přístupný přes 2 vodičovou sériovou sběrnici kompatibilní s I2C. Digitální teplotní senzor TC74 je dostupný v baleních SOT-23 a TO-220 obr.5.7.

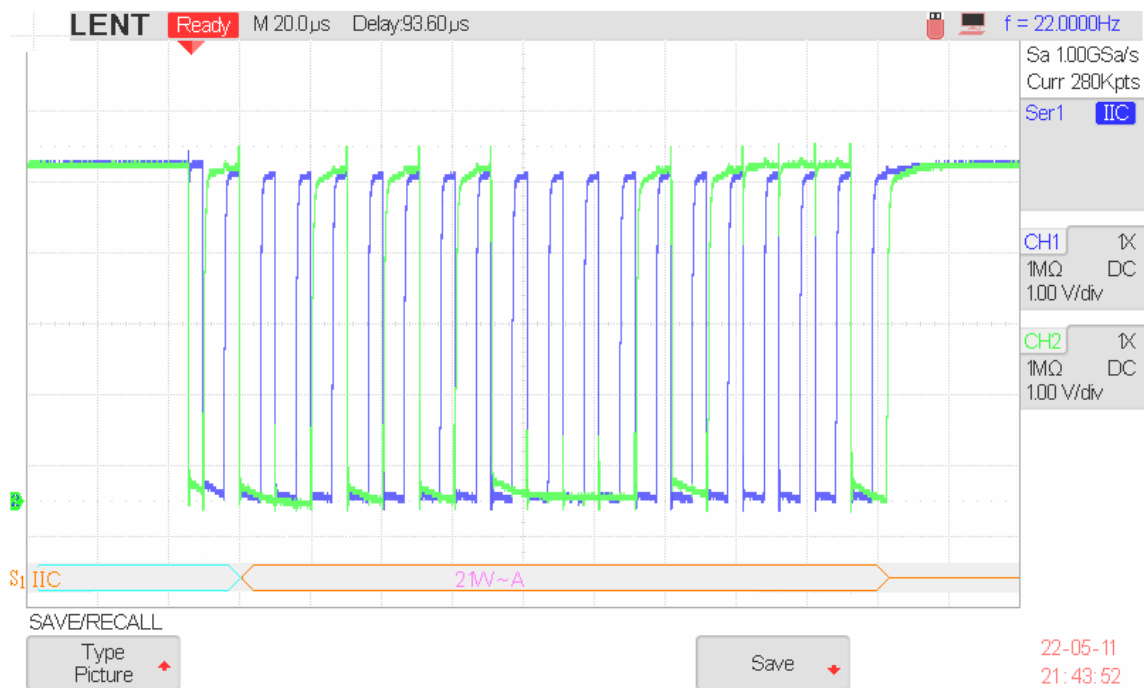


Obrázek 5.7: Provedení čipu TC74[34]

TC74 ukládá naměřenou teplotu do svého 8bitového interního registru jako binární formát komplementu 2. Nejvýznamnější bit je bit znaménka, který je nastaven na 1 pro záporné teploty. Maximální měřitelná kladná teplota je tedy $+127\text{ }^{\circ}\text{C}$ (0111 1111). TC74 má také další 8bitový registr pro čtení/zápis konfigurace (RWCR), který se používá k uvedení zařízení do pohotovostního režimu s nízkou spotřebou okolo $5\text{ }\mu\text{A}$. V tomto režimu je A/D převodník zastaven a registry teplotních dat jsou zmrazeny. Aby se TC74 přepnul do pohotovostního režimu, musí být nastaven bit 7 RWCR.[35]

5.4.1 Ukázka měření

V rámci testování byla měřena probíhající komunikace, a jak lze vidět na obrázku 5.8, komunikace probíhá dle teoretických předpokladů. V první části lze přečíst adresu senzoru a na konci odpověď v podobě teploty. Teplota je dle senzoru "0010 111" binárně, což odpovídá 23°C . Hrany signálu nejsou zcela kolmé a lze si všimnout drobných peaků. Tyto chyby však nejsou způsobeny na straně vyvíjeného zařízení, ale jsou zaviněny nedokonalým měřením, kdy byla do měřicího procesu zanesena parazitní indukce a kapacita.



Obrázek 5.8: Měřená probíhající komunikace I2C

5.5 Simulace SPI

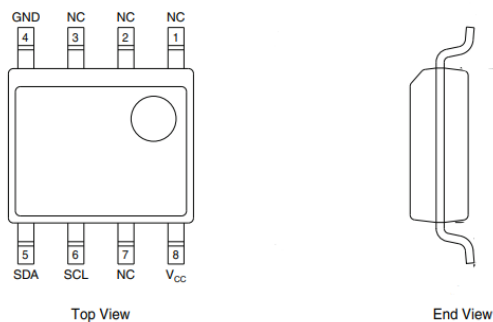
K účelům simulace je deska plošného spoje osazena pamětí EEPROM kompatibilní se sériovou komunikací SPI. Pro názornost komunikace je paměť spojena s mikrokontrolérem nejen datovými vodiči, ale také pomocnými, které by nebyly v případě jednoho slave zařízení na desce potřeba, avšak v případě více slave zařízení při běžném použití už ano.

Možnosti nastavení

- Rychlost komunikace 537,1 KB až 68,75 MB
- Adresa paměťového prostoru v rozmezí 1024 bitů
- Bytová délka čtených dat
- Bit CS

- Bit HOLD
- Bit WP

Paměť EEPROM AT25010B poskytuje 1024 bitů sériově elektricky vymazatelné programovatelné paměti organizované jako 128 slov po 8 bitech. Zařízení je optimalizováno pro použití v mnoha průmyslových a komerčních aplikacích, kde je nezbytný provoz s nízkým výkonem a nízkým napětím. AT25010B je k dispozici v prostоровě úsporných pouzdrech, ze kterých bylo využito JEDEC SOIC obr.5.9. Paměť AT25010B je vybrána prostřednictvím pinu Chip Select (CS), kdy je přístupná přes třívodičové rozhraní tvořené ze sériového datového vstupu (SI), sériového datového výstupu (SO) a sériových hodin (SCK). Všechny programovací cykly jsou zcela časově nezávislé a před zápisem není vyžadován žádný samostatný cyklus mazání. Bloková ochrana proti zápisu lze nastavit pomocí stavového registru, což je jednou ze možností ochrany proti přepisu. Pak také hardwarová ochrana dat je prostřednictvím pinu WP, který chrání před neúmyslnými pokusy o zápis a pin HOLD lze použít k pozastavení jakékoli sériové komunikace bez resetování sériové sekvence. [36]



Obrázek 5.9: Pouzdro paměti AT25010B-SSHL-B [36]

5.6 Paralelní sběrnice

Pro znázornění paralelní komunikace se používá 16x2 LCD displej, který využívá paralelní komunikaci při komunikaci z mikrokontrolérem. LCD displej obsahuje

ovladač HD44780. HD44780 je zodpovědný za veškerou komunikaci s externím ovladačem a zobrazování textu na LCD obrazovce. HD44780 podporuje 255 znaků, díky čemuž dokáže zobrazit 255 různých znaků na LCD přímo prostřednictvím ovladače. Podporované znaky jsou ASCII, číslice (0,9) a japonské znaky.[37] Pokud je potřeba zobrazit znak, vytvoří se na paralelní sběrnici adresa tohoto symbolu dle obrázku 5.10.

Lower 4 Bits	Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)	▶		0	1	2	3	4	5	6	7	8	9	A	B	C	D
xxxx0001	(2)	◀	!	1	A	Q	a	q	Ä	J	i	±	Á	Ñ	á	ñ	
xxxx0010	(3)	“	”	2	B	R	b	r	Ж	Г	Ф	²	À	Ò	á	ó	
xxxx0011	(4)	”	#	3	C	S	c	s	З	π	€	³	Ã	Ó	ä	õ	
xxxx0100	(5)	⌘	\$	4	D	T	d	t	Н	Σ	×	₣	Ä	Ö	ä	ö	
xxxx0101	(6)	₹	%	5	E	U	e	u	Й	σ	¥	₤	Å	Ö	ä	ö	
xxxx0110	(7)	⊙	&	6	F	V	f	v	И	Д	!	₧	Æ	Ö	æ	ö	
xxxx0111	(8)	⌘	'	7	G	W	g	w	П	τ	§	•	₪	×	₹	÷	
xxxx1000	(1)	↑	(8	H	X	h	x	У	*	₣	ω	È	Æ	è	⊙	
xxxx1001	(2)	↓)	9	I	Y	i	y	Ч	θ	⊙	¹	É	Ù	é	ù	
xxxx1010	(3)	→	*	:	J	Z	j	z	Ч	Ω	⊙	⊙	Ê	Ú	ê	ú	
xxxx1011	(4)	←	+	;	K	[k	[Ш	δ	⊙	⊙	Ë	Û	ë	û	
xxxx1100	(5)	≤	,	<	L	\	l		Щ	∞	⊙	⊙	Ï	Ü	ï	ü	
xxxx1101	(6)	≥	-	=	M]m]m]m	Ъ	⊙	⊙	⊙	İ	ÿ	ı	ÿ	
xxxx1110	(7)	▲	.	>	N	^	n	~	Ы	ε	⊙	⊙	İ	İ	ı	İ	
xxxx1111	(8)	▼	/	?	O	_	o	ˆ	Э	⊙	⊙	⊙	İ	İ	ı	İ	

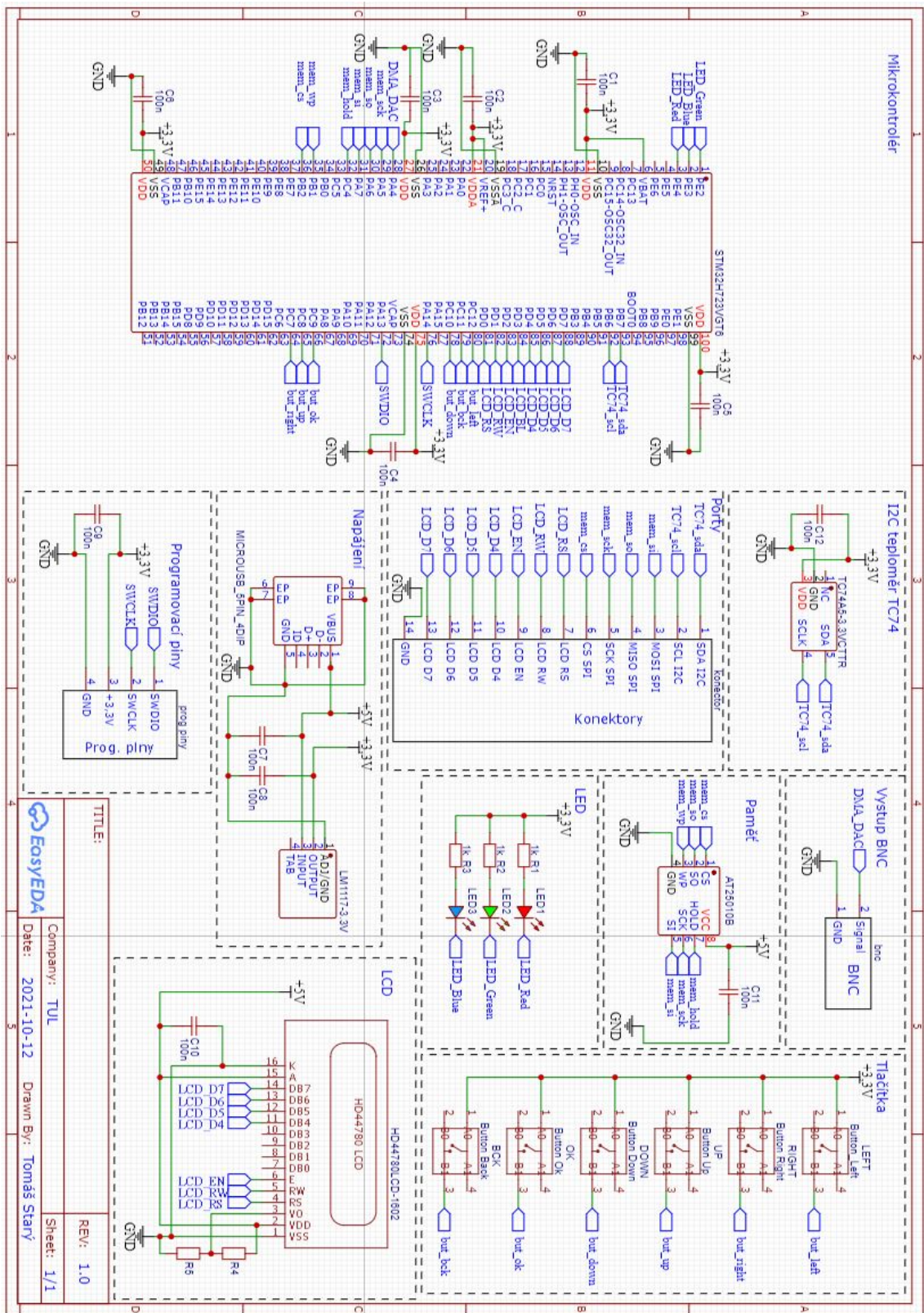
Obrázek 5.10: Adresa symbolů v ovladači HD44780[37]

Tímto způsobem funguje i napsaný program v mikrokontroléru. Při zobrazení například obrazovky s menu výběru signálu se napsaný text rozdělí na znaky. Jednotlivé znaky následně převedou na ASCII adresu. Z důvodu 4bitovému režimu se rozdělí ASCII adresa na horní 4 bity a spodní 4 bity, které následně pošlou po paralelní sběrnici do registru ovladače HD44780, který je zobrazí na displeji. Datové

linky mezi displejem a kontrolérem jsou zároveň vyvedeny na řadu pinů, na které se lze připojit pomocí paralelního rozhraní na osciloskopu a při správném nastavení je lze i dešifrovat a zobrazit na jeho obrazovce.

6 Návrh plošného spoje

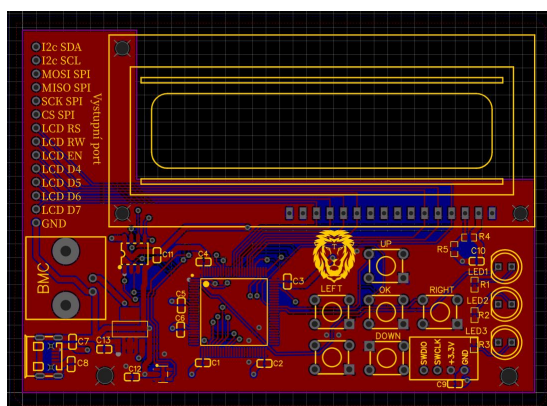
Pro osazení a spojení všech součástek je navržen a vyroben plošný spoj. Z důvodu práce s osciloskopem a následnou analýzou signálů byl při návrhu stínění a frekvenční nezávislosti plošného spoje kladen velký důraz na to, aby vnější signály neovlivňovaly jak samotný chod mikrokontroléru, tak generovaný signál pro následnou analýzu. Pro návrh bylo využito prostředí EasyEda z důvodu jeho rozsáhlých oficiálních knihoven výrobce, komunitní databázi sdílených návrhů komponent a v neposlední řadě i uživatelské přívětivosti. Jak lze na schématu plošného spoje na obrázku 6.1 vidět, na desce jsou dvě úrovně napájecího napětí. První je 5V z micro USB, sloužící k napájení LCD displeje s integrovaným obvodem HD44780 obr.4.13, paměťového čipu EEPROM AT25010B a napěťového regulátoru LM111, který z 5V vytvoří 3,3V. Napěťová úroveň 3,3V se následně využívá k napájení zbytku integrovaných obvodů včetně mikrokontroléru. Dále je plošný spoj osazen kondenzátory sloužícími k stabilizaci napájecího napětí a odporovým děličem napětí pro nastavení intenzity podsvícení LCD displeje, LED diodami pro signalizaci správného chodu programu, napájení desky a vytváření signálu na výstupu. Na plošném spoji se dále nachází tlačítka, která jsou rozmístěna do osového kříže a jedno samostatné v levém dolním rohu. Tato tlačítka slouží k orientaci v menu, ke změnám hodnot v nastavení, potvrzování výběru a návratu zpět do předchozího menu. Výstupní prvky jsou tvořeny BNC konektorem pro generované signály do osciloskopu, pinovým konektorem pro možnost čtení sériové komunikace I2c, SPI a dále paralelní komunikace mezi mikrokontrolérem a LCD displejem. Deska plošného spoje dále obsahuje čtyři piny pro programování mikrokontroléru, ty však jsou za normálních podmínek pod krytem a uživatel k nim nemá přístup.



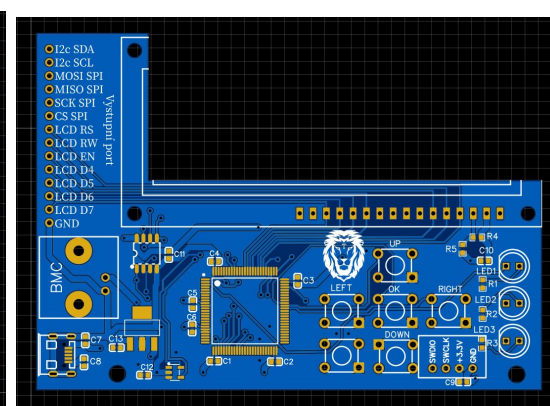
TITLE:
 Company: TUL
 Date: 2021-10-12
 Drawn By: Tomáš Stary
 REV: 1.0
 Sheet: 1/1

Obrázek 6.1: Schéma zapojení přípravku

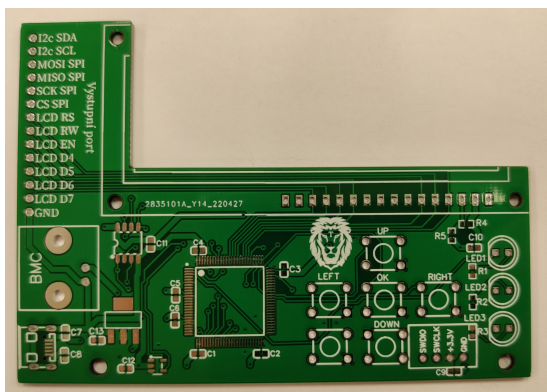
Deska plošného spoje obr. 6.2 je vytvořena tak, aby se nacházely všechny výstupy na levé straně. Od shora jsou piny pro sériovou, paralelní komunikaci a pod nimi konektor na BNC. Ve spodní části se nachází USB micro na napájení desky. Ovládací tlačítka tvoří kříž v pravém spodním rohu, vedle kterého se nachází piny pro programování a signalizační LED diody. Pro maximalizaci napájecích vodičů a snížení dopadu působení vnějších elektromagnetických polí byla použita rozlitá měď s potenciálem 3,3V a GND. V desce jsou vytvořeny otvory pro upevňovací šrouby M3. Ve spodní části šrouby připevní přímo desku plošného spoje, avšak uprostřed prochází zároveň deskou na LCD displeji a slouží k připevnění i jeho samotného.



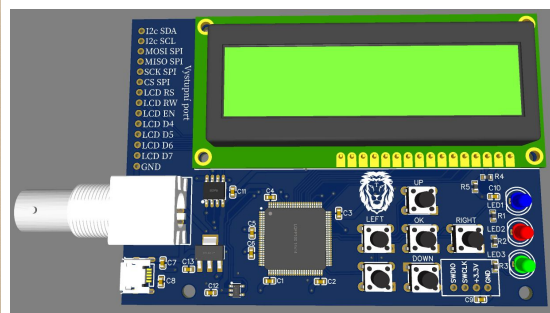
(a) Návrh desky plošného spoje



(b) 2D vizualizace plošného spoje



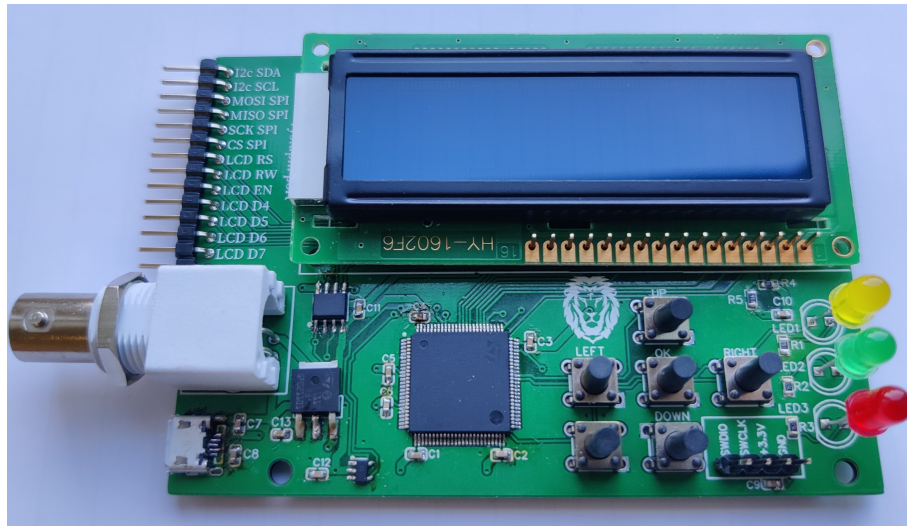
(c) 2D vizualizace plošného spoje



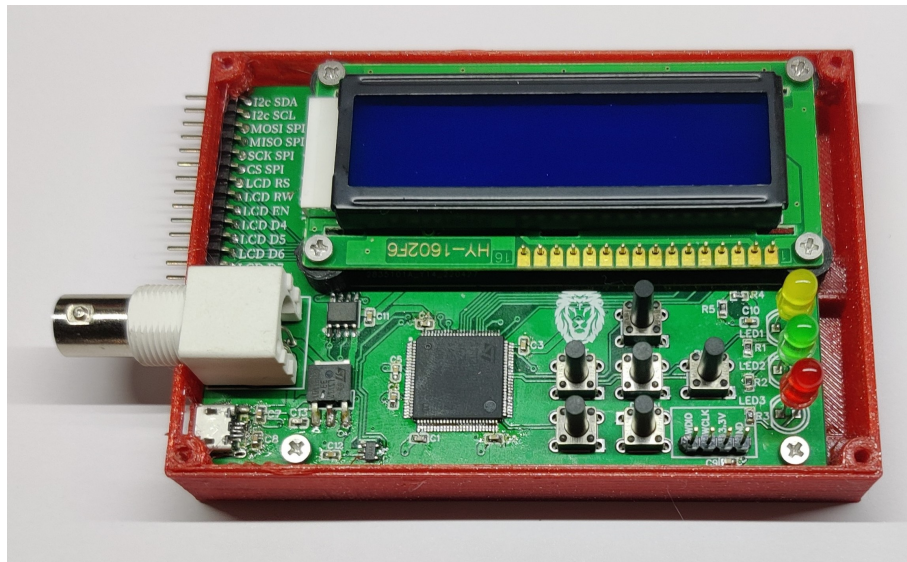
(d) 3D vizualizace osazeného plošného spoje

Obrázek 6.2: Návrh plošného spoje

Při výběru pouzder čipů byla preferována, z důvodu snadného osazení, pouzdra typu smd a umístění na vrchní straně plošného spoje.



Obrázek 6.3: Osazený plošný spoj



Obrázek 6.4: Osazený plošný spoj v krabičce

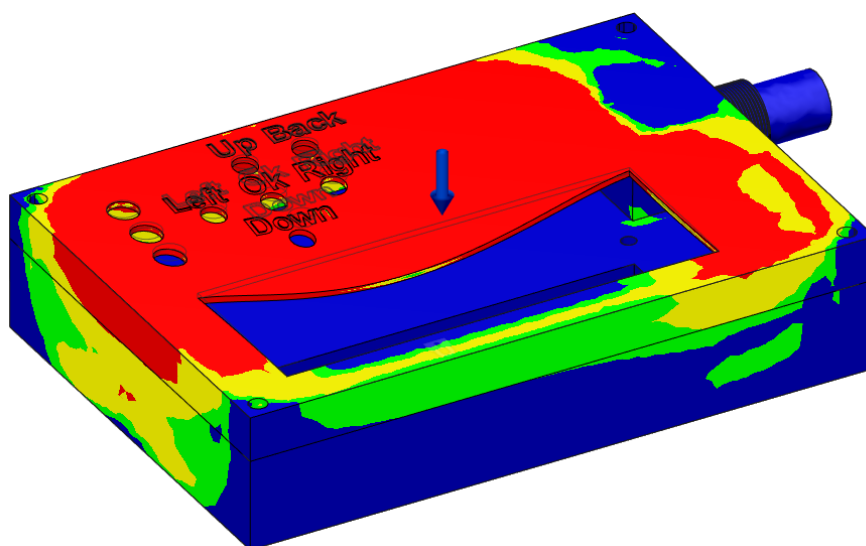
7 Návrh krabičky

Při návrhu krytu plošného spoje byl kladen důraz na jeho snadnou a rychlou vyrobiteľnost v podobě prototypů a na mechanickou odolnost. Z toho důvodu byl vybrán 3D tisk, který se díky své možnosti prototypové výroby stal ideální volbou. Po porovnání různých technologií byla vybrána technologie fused deposition modeling obr.7.1. FDM (Fused deposition modeling), je také známá jako Fused Filament Fabrication (FFF). Tato technologie je nejrozšířenějším typem 3D tisku na spotřebitelské úrovni. 3D tiskárny FDM pracují tak, že vytlačují termoplastická vlákna, jako je ABS (akrylonitril-butadienstyren), PLA (polylactic acid) či PETG (polyethylene terephthalate glycol) přes vyhřívanou trysku. Tryska taví a tak materiál nanáší vrstvu po vrstvě na stavební platformu. Vrstvy jsou pokládány na sebe, dokud není díl kompletní. Z výše uvedených materiálů byl vybrán PETG. Má oproti PLA větší pevnost a na rozdíl od ABS lze tisknout za standardních podmínek, kdy není potřeba řešit nebezpečné výpary při tisku.[38]



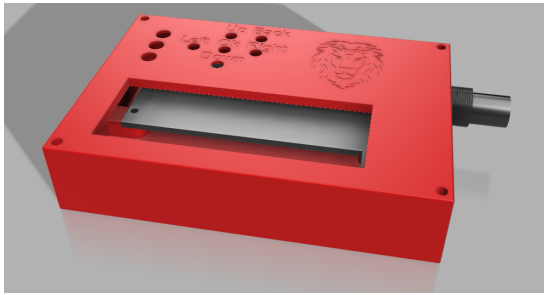
Obrázek 7.1: 3D tiskárna technologie FDM[39]

Při návrhu byl využit program Autodesk Fusion 360. Nejdříve byl zhotoven přesný model plošného spoje s otvory pro upevnění a modely konektorů. Následně by vytvořen první prototyp krytu. Rozměry byly přesně odvozeny od desky plošného spoje. Deska plošného spoje je přitom uchycena na válcové špalíky umístěné pod otvory skrze desku. Na uchycení byly v návrhu použity šrouby M2.5, kdy je v plastu krytu vymodelován závit. Kryt dále obsahuje ve vrchní straně otvory pro displej, tlačítka a diody, v boku krytu prostup na piny od komunikací, BNC konektoru a otvor pro zapojení microUSB. Na takto vymodelované krabičce byla provedena strukturální analýza. Jak je patrné na obr.7.2, model se při strukturální analýze za působení síly z vrchu prohýbá v slabém místě u hrany displeje, a proto bylo dále provedeno několik dalších optimalizací, kterými se mimo jiné zvýšila tuhost okraje u displeje.

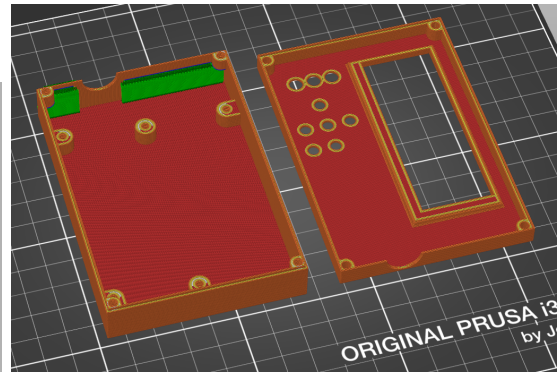


Obrázek 7.2: Kryt při strukturální analýze

Takto vytvořený návrh krytu obr.7.3(a) se v podobě STL modelu vložil do sliceru. Úkolem sliceru je převést 3D objekt z běžného 3D formátu do tisknutelné podoby, tj. rozřezat ho na vrstvy a naplánovat pohyby tiskové hlavy. Výsledný model obr.7.3(b) je doplněn o podpory v místech, kde by tisková vrstva neměla předchozí podkladní vrstvu a následně byl v podobě G-kódu nahrán do tiskárny.



(a) 3D vizualizace krytu



(b) Model v sliceru připravený k tisku

Obrázek 7.3: Kryt na zařízení



Obrázek 7.4: Hotová krabička s přípravkem uvnitř

8 Návrh laboratorní úlohy

Při výběru možných úloh, které by mohli studenti s přípravkem realizovat byla brána v úvahu hlavní tři kritéria.

- Rozšíření znalostí v práci s osciloskopem
- Propojení teoretických znalostí s praxí
- Zajímavost měření

Po rozboru možných funkcí bylo rozhodnuto o výběru simulování I2C komunikace. Rozšíření znalostí v práci s osciloskopem je zde zcela splněno. Díky nemožnosti využití záchranného tlačítka "auto setup", které by v případě jednoduchých signálů provedlo veškerou práci za studenta, je zde potřeba rozumět nejen samotné komunikaci, ale také možnostem triggru, zobrazení dvou signálů současně na jedné obrazovce a jejich vzájemné synchronizaci. Propojení teoretických znalostí s praxí přitom spočívá v práci s datasheetem. Studenti mohou znát z přednášky princip I2C komunikace, sled bitů a jejich délku, avšak je také třeba si umět zjistit konkrétní informace pro jednotlivá zařízení, jako je například adresa, podporované rychlosti přenosu, nebo jak lze přenesená data z této komunikace interpretovat. Je to jedním z důvodů proč se v této práci používá teploměr TC74. Jeho rozšířenost a kvalitně zpracovaná dokumentace nabízí výbornou příležitost naučit se pracovat s datasheety. Také snímaná teplota, zobrazená v binárním stavu, lze snadno ověřit. Student má při zobrazení komunikace dobrou představu, jakou hodnotu může přibližně čekat a proto si může snadno ověřit správné pochopení komunikace. Třetím kritériem je zajímavost měření. Toto kritérium se týká nejen originality, kdy měření není jen

všední měření signálu, například z dvojbranu, které je však také důležité, ale spočívá v plném využití zařízení, které je popisováno v této práci. Student může nastavovat parametry přenosu a pozorovat, co se stane v případě, že přesáhne podporované hodnoty. Má možnost díky teploměru uměle zvedat hodnoty posílané po komunikaci a živě sledovat jejich nárůst na komunikaci.

9 Závěr

Práce se zabývala vývojem zařízení, které dokáže nejen zlepšit kvalitu a efektivitu vzdělávání při práci s osciloskopem, ale také otestovat osciloskop samotný. Při vývoji zařízení byl kladen důraz na neovlivnitelnost generovaných funkcí změnami vnějšího prostředí. Proto jsou všechny vstupy funkcí, které generují signál, vytvořeny uvnitř zařízení a tím pádem jsou signály snadno replikovatelné. Díky tomu lze úlohy simulované zařízením následně porovnávat s kontrolním měřením při stejném nastavení, a dosáhnout tak totožného výsledku. To ulehčí kontrolu ze strany školitele, i školené obsluhy samotné. Velkou částí diplomové práce je také generování šumového signálu. Šum je základem měřicí techniky a jeho znalost při práci s osciloskopem stěžejní dovedností. Vyvinuté zařízení umožňuje generovat šum různých parametrů a různé četnosti výskytu. V moderních osciloskopech je také běžným standardem měření sériových i paralelních komunikací. U nich je náročná jejich nastavitelná simulace, a proto je jí v práci také věnována značná pozornost. Velkou výhodou vytvořeného zařízení je parametrizace, kdy lze měnit adresaci, rychlost přenosu či mnoho dalších parametrů a následně sledovat, jak se takto změněná komunikace bude chovat. Díky vnějším čipům na navržené desce plošného spoje je možné sledovat komunikaci v reálném čase i na fyzickém hardwaru, což je pro výuku velice důležité. Školené osoby mohou vykonávat stejný postup, jako by prováděly měření na chybové implementaci reálného zařízení a díky tomu se naučit, jak rozpoznat, co je na komunikaci chybně. Součástí práce bylo vytvoření schématu zapojení, dle kterého byl následně navržen a zhotoven plošný spoj. Plošný spoj je osazen mikrokontrolérem s řídicím programem, tlačítky, LCD displejem a kontrolními diodami. Celé zařízení bylo na závěr umístěno v navrženém a 3D vtištěném krytu, který ho chrání před poškozením.

Použitá literatura

- [1] *Oscilloscope: What is oscilloscope?* [Online]. 1999 [cit. 2022-03-19]. Dostupné z: www.techtarget.com/whatis/definition/oscilloscope.
- [2] *Cathode-Ray Oscilloscope* [online] [cit. 2022-04-12]. Dostupné z: <http://boson.physics.sc.edu/~hoskins/Demos/CathodeRay.html>.
- [3] HAVLÍK, Ladislav. *Oscilloskopy a jejich použití*. Praha: Sdělovací technika, 2002. ISBN 80-901-9368-4.
- [4] MATĚJKOVÁ, Helena. *Měření kmitočtu a tvaru signálů pomocí osciloskopu* [online]. ADOC.PUB [cit. 2022-01-08]. Dostupné z: www.adoc.pub/meni-kmitotu-a-tvaru-signal-pomoci-osciloskopu.html.
- [5] *Digital Oscilloscope or (DSO): What is it and how is it used?* [Online] [cit. 2022-04-03]. Dostupné z: www.testandmeasurementtips.com/what-is-a-digital-oscilloscope/.
- [6] *User Manual: SDS2000 Series Digital oscilloscope* [online] [cit. 2022-01-11]. Dostupné z: https://siglentna.com/USA_website_2014/Documents/UserManual/SDS2000_UserManual_UM01020-E03A.pdf.
- [7] ROBENEK, Jan. *Z(a)traceno v šumu* [online]. 1997: HW server s.r.o. [cit. 2022-02-17]. Dostupné z: www.vyvoj.hw.cz/teorie-a-praxe/zatraceno-v-sumu.html.
- [8] PINI, ARTHUR. *Measure vector and area with an oscilloscope X-Y display* [online] [cit. 2022-04-16]. Dostupné z: www.edn.com/measure-vector-and-area-with-an-oscilloscope-x-y-display/.

- [9] HERRES, DAVID. *The basics of digital oscilloscope acquisition controls* [online] [cit. 2022-05-08]. Dostupné z: www.testandmeasurementtips.com/basics-digital-oscilloscope-acquisition-controls.
- [10] DATASHEET. *Segmented Memory Acquisition for InfiniiVision Series Oscilloscopes* [online]. Keysight Technologies [cit. 2022-02-02]. Dostupné z: www.keysight.com/us/en/assets/7018-01736/data-sheets/5989-7833.pdf.
- [11] *STM32F030R8T6: ARM MCU, Value Řada, STM32 Family STM32F0 Series Microcontrollers, ARM Cortex-M0, 32bitů, 48 MHz* [online] [cit. 2022-06-01]. Dostupné z: www.cz.farnell.com/stmicroelectronics/stm32f030r8t6/mcu-32bit-cortex-m0-48mhz-lqfp/dp/2393636.
- [12] MAGDY, KHALED. *STM32 DAC Tutorial – Example HAL Code & Analog Signal Genreation: STM32 DAC Tutorial With Examples To Generate Analog Signals* [online] [cit. 2022-02-14]. Dostupné z: <https://deepbluembedded.com/stm32-dac-tutorial-example-hal-code-analog-signal-genreation/>.
- [13] *STM32H7 Series* [online] [cit. 2022-04-08]. Dostupné z: www.st.com/en/microcontrollers-microprocessors/stm32h7-series.html%5C#overview.
- [14] *NUCLEO-H743ZI* [online] [cit. 2022-01-12]. Dostupné z: www.os.mbed.com/platforms/ST-Nucleo-H743ZI.
- [15] ASADI, Farzin a Kei EGUCHI. *Electronic Measurements: A Practical Approach*. Morgan & Claypool Publishers, 28. 2. 2021. ISBN 13: 978-1636390925.
- [16] PAONESSA, Simon. *Reducing Signal Noise in Practice* [online]. 233 South St, Hopkinton, MA 01748, Spojené státy: Precision Digital Corporation [cit. 2022-03-07]. Dostupné z: www.predig.com/whitepaper/reducing-signal-noise-practice.
- [17] *How to reduce signal noise on a basic oscilloscope* [online] [cit. 2022-01-06]. Dostupné z: <https://www.tek.com/en/blog/how-reduce-signal-noise-basic-oscilloscope>.

- [18] Chapter 2: Statistics, Probability and Noise. In: *The Scientist and Engineer's Guide to Digital Signal Processing*. 1st edition. California Technical Pub, [b.r.]. ISBN 978-0966017632.
- [19] ABDIPOUR, A., G. MORADI a S. SABOKTAKIN. Design and implementation of a noise generator. *2008 IEEE International RF and Microwave Conference* [online]. 2008, s. 472–474 [cit. 2022-02-06]. ISBN 978-1-4244-2866-3. Dostupné z DOI: [10.1109/RFM.2008.4897418](https://doi.org/10.1109/RFM.2008.4897418).
- [20] TA, -. *A Tube Based Noise Generator: Some Noise Theory* [online]. Graz, Austria: FH JOANNEUM, 2019 [cit. 2022-01-02]. Dostupné z: www.ta.linzbichler.net/en/posts/audio/a-tube-based-noise-generator-2/.
- [21] HOROWITZ, Paul a Winfield HILL. *The art of electronics*. 2nd ed. Cambridge: University Press, 1989. ISBN 05-213-7095-7.
- [22] CHRISTOFFERSEN, Jens. *Switch Bounce and How to Deal with It* [online] [cit. 2022-04-01]. Dostupné z: www.allaboutcircuits.com/technical-articles/switch-bounce-how-to-deal-with-it/.
- [23] *Relay contact bounce test* [online] [cit. 2022-04-19]. Dostupné z: <http://electromotiveforces.blogspot.com/2011/01/relay-contact-bounce-test.html>.
- [24] MAXFIELD, Clive. *How to Implement Hardware Debounce for Switches and Relays When Software Debounce Isn't Appropriate* [online]. Contributed By Digi-Key's North American Editors [cit. 2022-01-08]. Dostupné z: <https://www.digikey.com/en/articles/how-to-implement-hardware-debounce-for-switches-and-relays>.
- [25] NEJEDLÝ, Tomáš Bc. *Dvojité inverzní kyvadlo na vozíku*. Technická univerzita v Liberci, 2022. Diplomová práce. Technická univerzita v Liberci.
- [26] MISHRA, Sanjeeb, Neeraj Kumar SINGH a Vijayakrishnan ROUSSEAU. Sensor Interfaces. *System on Chip Interfaces for Low Power Design* [online]. 2016, s. 331–344 [cit. 2022-03-03]. ISBN 9780128016305. Dostupné z DOI: [10.1016/B978-0-12-801630-5.00010-4](https://doi.org/10.1016/B978-0-12-801630-5.00010-4).

- [27] DAVIS, ZIFF. *EEPROM* [online]. PCMAG DIGITAL GROUP, 1996 [cit. 2022-04-09]. Dostupné z: www.pcmag.com/encyclopedia/term/eprom.
- [28] PELIKÁN, Jaroslav. *Vnitřní paměti* [online]. Masarykova univerzita Fakulta informatiky, 1999 [cit. 2022-05-01]. Dostupné z: <https://www.fi.muni.cz/usr/pelikan/ARCHIT/TEXTY/INTPAM.HTML>.
- [29] RATNA. *Parallel Communication – Evolution, Works, Characteristics, When to Use* [online] [cit. 2022-01-02].
- [30] *1602 SMART LCD DISPLAY MODULE* [online]. Hobbycomponents.com [cit. 2022-04-02]. Dostupné z: http://hobbycomponents.com/downloads/HCMODU0122_1602_Smart_LCD_Manual.pdf.
- [31] *LCD technology* [online]. Xenarc Technologies Corp [cit. 2022-03-02]. Dostupné z: www.xenarc.com/lcd-technology.html.
- [32] GIOVANNI, Nicholas Di a Cedric DEMERS. *Plasma vs LCD & LED How They Work* [online]. 160 St-Viateur Est, Suite 602, Montreal, Canada: Quebec Inc [cit. 2022-03-12]. Dostupné z: www.rtings.com/tv/learn/lcd-vs-led-vs-plasma/how-they-work.
- [33] *STM32H7-Security-Random Number Generator* [online]. STMicroelectronics [cit. 2022-02-08]. Dostupné z: https://www.st.com/content/ccc/resource/training/technical/product_training/group0/5a/ba/43/28/46/d0/44/aa/STM32H7-Security-Random_Number_Generator_RNG/files/STM32H7-Security-Random_Number_Generator_RNG.pdf/_jcr_content/translations/en.STM32H7-Security-Random_Number_Generator_RNG.pdf.
- [34] DATASHEET. *TC74: Tiny Serial Digital Thermal Sensor* [online]. Microchip Technology Inc., 2002 [cit. 2022-04-14].
- [35] R-B. *USING TC74 THERMAL SENSOR FOR TEMPERATURE MEASUREMENT* [online] [cit. 2022-03-15]. Dostupné z: www.embedded-lab.com/blog/using-tc74-microchip-thermal-sensor-for-temperature-measurement/.

- [36] *SPI Serial EEPROM: Atmel AT25010B* [online]. Atmel Corporation [cit. 2022-03-19]. Dostupné z: [www . tme . eu / Document / d0654aeab010c376db857072bea94057/doc8707.pdf](http://www.tme.eu/Document/d0654aeab010c376db857072bea94057/doc8707.pdf).
- [37] DATASHEET. *HD44780U (LCD-II)* [online]. Hitachi Semiconductor [cit. 2022-03-01]. Dostupné z: www.pdf1.alldatasheet.com/datasheet-pdf/view/63673/HITACHI/HD44780.html.
- [38] *FDM vs. SLA: Compare Filament and Resin 3D Printers* [online] [cit. 2022-04-02]. Dostupné z: www.formlabs.com/blog/fdm-vs-sla-compare-types-of-3d-printers/.
- [39] *3D tiskárna Original Prusa i3 MK3S+* [online] [cit. 2022-05-01]. Dostupné z: www.prusa3d.com/cs/produkt/3d-tiskarna-original-prusa-i3-mk3s-3/.

A Přílohy

- Zadání úlohy pro laboratorní úlohu s využitím navrženého přípravku.
- Programové soubory do testovacího zařízení
- Model krabičky ve formě STL souboru.



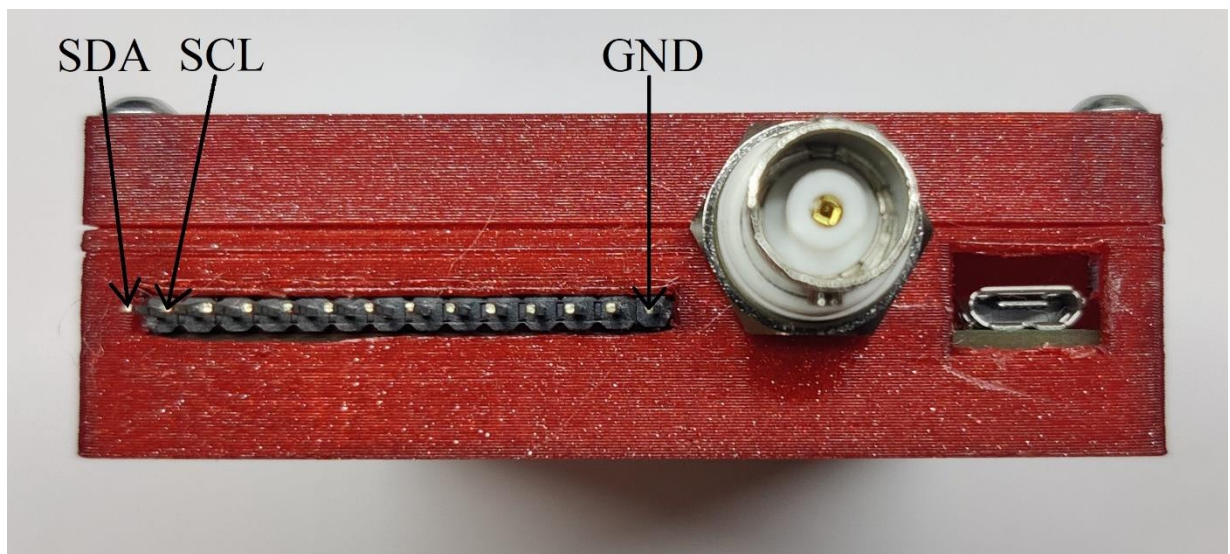
Komunikace I2C

Měření a diagnostika komunikace I2C

1. *Práce s datasheetem a nastavení komunikace*
2. *Diagnostika a čtení bitů na komunikaci pomocí osciloskopu*

Postup měření

1. Zapojení a nastavení osciloskopu
 - a. Podle obrázku spojte CH1 na osciloskopu s pinem SCL (druhy pin od spodu) a CH2 s SDA (první pin od spodu). U obou kanálů propojte referenční země s pinem GND (vrchní pin).



- b. Stiskněte **Decode** na pravé straně osciloskopu, vyberte **Serial 1** a komunikaci **I2C**.
- c. Následně se v **Signal** menu pod tlačítkem **Decode** přesvědčte, že je správně nastavená komunikace **SCL** na **CH1** a **SDA** na **CH2**. Úroveň **Threshold** nastavte na hladinu 1,6V.



- d. Pod tlačítkem **Setup** v nastavení triggeru vyberte **Serial 1 I2C** a **Start** v nabídce **Condition**.





2. Výukové zařízení je osazeno teplotním senzorem **TC74A2-3.3VCT**. V datasheetu tohoto senzoru zjistěte adresu, komunikační rychlost, počet bajtů čtených dat a vše nastavte pomocí tlačítek do nastavení komunikace I2C na výukovém zařízení.
3. Zaznamenejte bitový přenos na osciloskopu, který následně popište. Rozdělte jej na Adresu a Data, označte v přenosu Startovací a Ukončovací podmínku, RD/WR bit, ACK bit a NACK bit.

Receive Byte Format

S	Address	RD	ACK	Data	NACK	P
	7 Bits			8 Bits		

4. Za pomoci datasheetu zjistěte z datového bajtu přenosu teplotu měřenou senzorem.
5. V nastavení výukového zařízení změňte přenosovou rychlost, adresu slave zařízení a popište změnu zobrazenou na osciloskopu.

