



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

ANALYZÁTOR PROTOKOLU DMX512 PRO OSOBNÍ POČÍTAČ

DMX 512 PROTOCOL ANALYZER FOR PERSONAL COMPUTER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Vilém Brouček

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Ondřej Krajsa, Ph.D.

BRNO 2024

Bakalářská práce

bakalářský studijní program **Audio inženýrství**
specializace Zvuková produkce a nahrávání
Ústav telekomunikací

Student: Vilém Brouček

ID: 240141

Ročník: 3

Akademický rok: 2023/24

NÁZEV TÉMATU:

Analyzátor protokolu DMX512 pro osobní počítač

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je návrh a realizace USB - DMX512 převodníku a vhodného software, který by zobrazoval provoz na DMX rozhraní a umožňoval jeho analýzu s možností generování DMX zpráv.

DOPORUČENÁ LITERATURA:

- [1] HOWELL, Wayne. Control Freak: a real world guide to DMX512 and Remote Device Management. Cambridge: Entertainment Technology Press, 2008, 226 s. : il. ISBN 978-1-904031-55-0
- [2] MAZIDI, Muhammad Ali, Sarmad NAIMI a Sepehr NAIMI. The AVR microcontroller and embedded systems: using Assembly and C. Second edition. 2017. ISBN 9780997925968.

Termín zadání: 5.2.2024

Termín odevzdání: 28.5.2024

Vedoucí práce: Ing. Ondřej Krajsa, Ph.D.

doc. Ing. Jiří Schimmel, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

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

ABSTRAKT

Tato bakalářská práce se zabývá návrhem a konstrukcí převodníku USB–DMX512. V práci je podrobně popsána funkce protokolu DMX512, zejména struktura jeho signálu a fyzické vrstvy. Dále jsou diskutovány možnosti analýzy signálu, je vybrán vhodný mikrokontrolér a vytvořen návrh převodníku.

V práci je zařízení zkonstruováno, jsou vysvětleny jeho funkce, vysvětlen výběr součástek a princip návrhu desky plošných spojů. Zároveň je zařízení popsáno z uživatelského hlediska, zmiňují se možná vylepšení pro další verze.

Práce se věnuje také návrhu kódu pro mikrokontrolér STM32, zejména tvorbě vlastních knihoven k analýze a generování signálu. Podobně je zaměřena na tvorbu aplikace pro osobní počítač, zdůvodňuje volbu programovacího jazyka C# a popisuje objektově orientované členění kódu, včetně vlastních knihoven pro generování signálu DMX512 a příjem dat z analyzátoru.

Výstupem práce je funkční analyzátor DMX512, který tento signál dokáže přijímat i odesílat, zobrazuje jeho hodnoty a umožňuje propojení s PC pomocí vytvořené aplikace.

KLÍČOVÁ SLOVA

Analyzátor, C#, DMX512, jevištní osvětlení, řízení scénické techniky, STM32, světelná technika, WPF

ABSTRACT

This bachelor thesis consists of designing and assembling an USB–DMX512 analyzer. It contains detailed information about the protocol itself, particularly the structure of the signal and its physical layer. It discusses options for analyzing the signal, chooses an appropriate microcontroller unit and provides a draft of a hardware for the controller. The thesis describes the assembled analyzer, explains its functions, the selection of components, and the principles of PCB design. It also explains the usage of the device and its potential future upgrades.

The thesis provides a description of the code for the STM32 microcontroller and the custom-made libraries used for the DMX512 packet analysis and creation. It then familiarizes the reader with the code of the personal computer application, the reasons for choosing C# as the main programming language, and explains the structure of the object-oriented code, along with the custom-made libraries for DMX512 packet generation and analysis of the received data.

The result of the thesis is a working DMX512 analyzer, which can receive and transmit the signal, display transmitted data, and connect to a PC via the developed software.

KEYWORDS

Analyzer, C#, DMX512, lighting technology, stage lightning, stage lighting control, STM32, WPF

BROUČEK, Vilém. *Analyzátor protokolu DMX512 pro osobní počítač*. Bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2024. Vedoucí práce: Ing. Onřej Krajsa, PhD.

Prohlášení autora o původnosti díla

Jméno a příjmení autora: Vilém Brouček
VUT ID autora: 240141
Typ práce: Bakalářská práce
Akademický rok: 2023/24
Téma závěrečné práce: Analyzátor protokolu DMX512 pro osobní počítač

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno

.....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Ondřeji Krajsovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci, a svým blízkým za pevnou podporu při studiu.

Obsah

Úvod	11
1 DMX512	12
1.1 Způsob komunikace	12
1.1.1 Slot	12
1.1.2 Adresace zařízení	13
1.2 Fyzická vrstva	15
1.2.1 Topologie	16
1.2.2 Používané konektory	16
1.2.3 Kabely a terminace signálu	17
1.2.4 Elektrická konstrukce – izolace, zemnění zařízení	18
1.2.5 Přídavné funkce protokolu	18
1.3 Komerčně dostupné analyzátoři	19
2 Návrh a realizace prototypu	20
2.1 Výběr vhodného mikrokontroléru	20
2.1.1 ESP32	21
2.1.2 STM32	22
2.1.3 Volba a porovnání	23
2.2 Realizace prototypu	25
3 Realizace analyzátoři	28
3.1 Funkce analyzátoři a jejich řešení	29
3.1.1 Analýza signálu	29
3.1.2 Generování	30
3.1.3 Vlastní protokol pro komunikaci	31
3.2 Popis hardware analyzátoři	33
3.2.1 Mikrokontrolér	33
3.2.2 Převodník RS485–TTL	34
3.2.3 Převodník USB–Serial	34
3.2.4 Napájení, napěťový regulátor LDO, spotřeba	35
3.2.5 LCD displej	35
3.2.6 Relé	35
3.2.7 Externí konektory	36
3.2.8 Interní konektory	36
3.2.9 Informační LED diody	37
3.2.10 Krabice	37

3.3	Konstrukce desky plošných spojů	37
3.4	Obsluha zařízení	39
3.5	Doporučené úpravy pro další verze	41
4	Návrh a realizace programu mikrokontroléru	44
4.1	Nastavení mikrokontroléru a periférií	44
4.2	Knihovna pro DMX512	44
4.3	Knihovna dotykového displeje	47
4.4	Grafické uživatelské rozhraní zařízení	47
4.5	Možné rozšíření SW analyzátoru	48
5	Návrh a realizace aplikace pro PC	49
5.1	Výběr programovacího jazyka	49
5.2	Výběr grafického rozhraní	49
5.3	Popis aplikace	50
5.4	Vnitřní struktura aplikace	52
5.4.1	Přijímání zpráv	52
5.4.2	Odesílání zpráv	54
5.4.3	Obsluha GUI	54
	Závěr	57
	Literatura	58
	Seznam symbolů a zkratk	62
	Seznam příloh	64
	A Obsah elektronické přílohy	65
	B Vnitřní uspořádání analyzátoru	66
	C Schéma analyzátoru	67
	D DPS analyzátoru	70

Seznam obrázků

1.1	Schéma paketů DMX512	13
1.2	Příklad topologie sítě DMX512	16
2.1	Prototyp analyzátoru	26
2.2	Blokové schéma prototypu	26
3.1	Zkonstruovaný analyzátor v krabičce	28
3.2	Blokové schéma analyzátoru	29
3.3	Analyzátořem generovaný signál – začátek paketu	30
3.4	Samostatné použití SW pro generování zpráv	31
3.5	Osazená DPS analyzátoru s popisky konektorů	33
3.6	Transily na ochranu proti přepětí	39
3.7	Hlavní menu analyzátoru	40
3.8	Režim manuální analýzy.	40
3.9	XLR konektory analyzátoru	41
4.1	Nastavení vývodů mikrokontroléřu v CubeMX	45
4.2	Vývojový diagram odesílání DMX512 z analyzátoru	46
5.1	Aplikace analyzátoru pro osobní počítač, rozložení matice kanálů	51
5.2	Aplikace analyzátoru pro osobní počítač, rozložení jednotlivých kanálů	52
5.3	Vývojový diagram přijímacího algoritmu v aplikaci	53
5.4	Vývojový diagram odesílacího algoritmu v aplikaci	55
B.1	Analyzátor bez víka	66
C.1	Schéma analyzátoru, list 1	67
C.2	Schéma analyzátoru, list 2	68
C.3	Schéma použitého relé	69
D.1	DPS – celkový náhled bez zobrazení GND	70
D.2	DPS – vrchní strana, cesty	71
D.3	DPS – spodní strana, cesty	71
D.4	DPS – vrchní strana, potisk	72

Seznam tabulek

1.1	Časování paketů DMX512	14
1.2	Seznam normou podporovaných Start kódů	14
1.3	Pinout pro konektor 5-pin XLR	17
1.4	Pinout pro konektor IEC 60603-7	17
1.5	Pinout pro konektor 3-pin XLR	18
1.6	Tabulka doporučených řešení zemnění	18
3.1	Formát vlastního protokolu pro odesílání	32
3.2	Funkce analyzátoru dle zvoleného režimu	41

Úvod

Řízení světelné techniky je již několik desítek let aktivně využíváno během živých uměleckých vystoupení k vykreslení atmosféry, vytvoření intenzivnějšího zážitku nebo jako výrazový prostředek. Při malých i velkých vystoupeních se využívá tvorby programů, které následně v reálném čase pomocí protokolu (např. MIDI, ACN nebo DMX512) ovládají až stovky propojených světelných zařízení. Nejčastěji mění intenzitu, barvu nebo směr vyzařovaného světla [1]. Řiditelné světelné obvody se však běžně objevují i mimo hudební či divadelní vystoupení. Často se s nimi setkáváme v domácnostech jako s dekorativními, ale i primárními zdroji světla. Využití nalézají při osvětlování exteriérů budov a památek nebo v inteligentním pouličním osvětlení. Jejich cíl však zůstává převážně stejný – poskytnutí světla s parametry ovladatelnými dle požadavků uživatele [2].

Tato práce se věnuje návrhu a realizaci analyzátoru protokolu DMX512 a vhodného software, který bude zobrazovat provoz na rozhraní DMX512 a umožňovat generování zpráv. V rámci této bakalářské práce bude proveden návrh zařízení a realizován jeho hardware včetně aplikace pro zachytávání a odesílání zpráv. Výsledné zařízení bude sloužit jako převodník, připojený k osobnímu počítači pomocí USB (Universal Serial Bus) ale bude umožňovat i stand-alone režim.

V teoretické části se bude práce věnovat podrobným specifikacím a funkcím protokolu DMX512. V další části bude vybírán vhodný mikrokontrolér a následně bude představen návrh zařízení a jeho prototyp.

V kapitole 2.2 bude představeno vytvořené zařízení, vysvětleny jeho funkce, představena deska plošných spojů a odůvodněn výběr součástek. V této kapitole bude také popsána obsluha zařízení a nastíněny úpravy pro jeho další verze.

V kapitolách 4 a 5 bude blíže představen koncept softwarové části práce a tvorba knihoven pro analýzu signálu. Bude vysvětlena základní funkce programu mikrokontroléru i aplikace pro osobní počítač.

1 DMX512

DMX512 je digitální sériový asynchronní protokol, který se používá pro vzdálené ovládání jevištní a scénické, především světelné techniky. Byl navržen v roce 1986 institutem USITT (United States Institute for Theatre Technology) jako náhrada za analogové řízení a dnes je standardizovaný dle ANSI (American National Standards Institute) v dokumentu *E1.11 – 2008, USITT DMX512-A* [3].

Typické použití protokolu zahrnuje světelný pult nebo počítač, jakožto zdroj signálu DMX512, a přijímače signálu – typicky otočné hlavy, stroboskopy, lasery, výrobny mlhy a podobné světelné efekty a zařízení. Mixážní pulsy i počítačové aplikace umožňují jak manuální řízení této techniky, tak automatické – nejčastěji pomocí sekvencí DMX512 příkazů. Ty lze vytvářet, upravovat a následně samostatně vyvolávat, nebo je používat v rámci větší sekvence napříč celou světelnou show. Při vystoupení tak lze řízenou techniku synchronizovat s ostatním multimediálním obsahem, jako například s vizualizací na velkoplošných obrazovkách, zvukem nebo pyrotechnikou.

1.1 Způsob komunikace

Princip komunikace protokolu DMX512 lze rozdělit do tzv. paketů [3, 4]. Začátek každého paketu označuje „break“ (pauza). Po dobu jejího trvání, nejméně 88 μ s (viz tabulku 1.1), je na sběrnici vysílána logická 0. Následuje prodleva „Mark" after break“ – na výstup sběrnice je vysílána logická 1, tzv. „marking state“ [3, 5]. Poté již následuje i první odesílaný byte, tzv. „slot“ (viz 1.1.1). První odesílaný slot se nazývá „Start code“ a určuje funkci všech následujících datových slotů, tzv. kanálů. Pro ovládání scénické techniky se využívá hodnota Start code 0x00. V tabulce 1.2 jsou následně popsány další použitelné či rezervované hodnoty. Po prvním slotu následuje dalších 512 datových slotů, které již nesou uživatelská data. Z toho také vychází název **DMX512**. Mezi jednotlivými datovými sloty může být až 1 s prodleva v marking state. Podobně je tomu i na konci jednotlivých paketů [3].

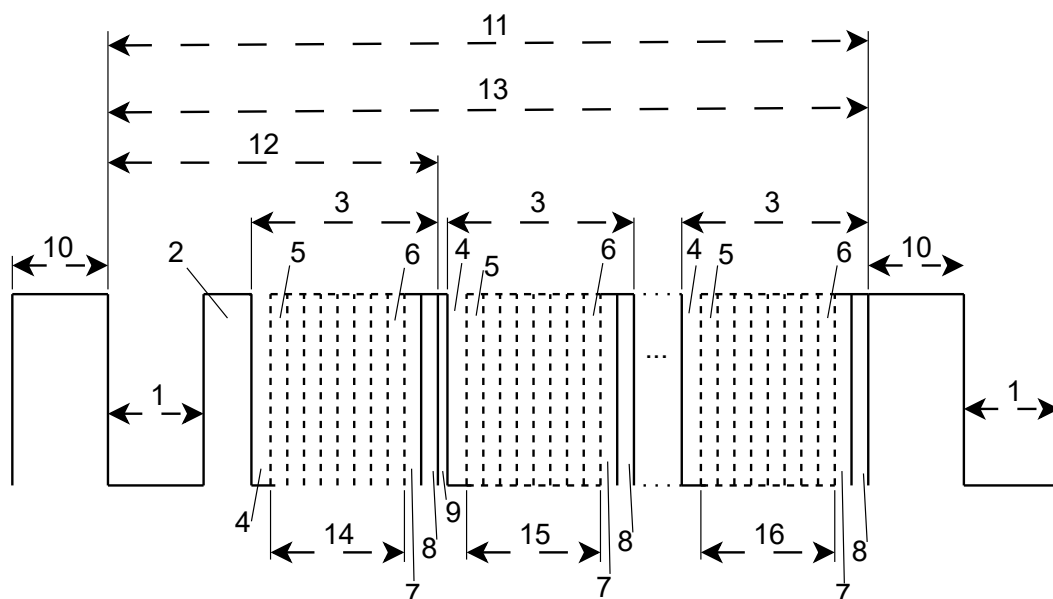
1.1.1 Slot

V protokolu DMX512 se jako „slot“ označují datové byty (viz obr. 1.1 a tab. 1.1). Ty vycházejí ze standardní sériové komunikace pomocí UART (Universal Asynchronous Receiver-Transmitter) a mají parametry [6]

- baud rate: 250 kbit/s,
- počet start bitů: 1,
- počet datových bitů: 8 (LSB první),
- počet stop bitů: 2,

- žádný paritní bit.

Z těchto parametrů vyplývá, že rozsah hodnot každého slotu (kanálu) je 2^8 , tedy 0–255.



Obr. 1.1: Schéma paketů v protokolu DMX512.

1.1.2 Adresace zařízení

Každé zařízení, které je připojeno ke sběrnici signálu přijímá vždy celý paket signálu. Většina běžných zařízení však napříč tomu pracuje kromě slotu *Start code* pouze s několika určitými datovými sloty v paketu. U světelné techniky se často setkáváme s 1 či 2kanálovými zařízeními – typicky divadelní osvětlení se stmíváním, s 4 až 8kanálovými zařízeními – typicky RGB světla. Otočné světelné hlavy pak často disponují 8 i více kanály. Počet kanálů reprezentuje množství po sobě jdoucích datových slotů, kterými může být zařízení ovládáno. Na které konkrétní kanály (datové sloty) bude zařízení reagovat, určuje jeho adresa.

Adresa musí být nastavena na každém zařízení, které má na sběrnici DMX512 přijímat data, a může nabývat hodnoty 1–512. Zvolená adresa představuje číslo kanálu (slotu), na jehož přenášené hodnoty bude zařízení reagovat. V případě, že je zařízení n -kanálové, bude reagovat i na $n - 1$ bezprostředně následujících kanálů (slotů). Adresy jsou volitelné zpravidla pomocí tlačítek nebo přepínačů na přijímacím zařízení a funkce přijímaných hodnot je pro jednotlivé kanály popsána v uživatelské příručce daného zařízení.

Tab. 1.1: Časování paketů DMX512 [3].

Č.	Popis	Min.	Typická	Max.
1	"SPACE" for BREAK	88 μ s	176 μ s	–
2	"MARK" After BREAK (MAB)	8 μ s	–	1,00 s
3	Slot Time	–	–	–
4	Start Bit	–	–	–
5	LEAST SIGNIFICANT Data BIT	–	–	–
6	MOST SIGNIFICANT Data BIT	–	–	–
7	STOP Bit	–	–	–
8	START Bit	–	–	–
9	"MARK" Time between slots	0	–	1,00 s
10	"MARK" Before BREAK (MBB)	0	–	1,00 s
11	BREAK to BREAK Time	1196 μ s	–	1,25 s
12	RESET Sequence	–	–	–
13	DMX512 Paket	1196 μ s	–	1,25 s
14	START CODE (SLOT 0 Data)	–	–	–
15	SLOT 1 DATA	–	–	–
16	SLOT nnn DATA (Maximum 512)	–	–	–

Tab. 1.2: Seznam normou podporovaných Start kódů a jejich funkce [3].

START Code		Funkce paketu
Hexadecimální	Dekadický	
00	0	Standardní způsob přenosu dat
17	23	Textový paket formátu ASCII
55	85	Označení testovacího paketu
90	144	Textový paket formátu UTF-8
91	145	Rezervováno pro potřeby organizací a výrobců
92–A9	146–169	Pro možné budoucí využití
AB–CD	171–205	Pro možné budoucí využití
CF	207	Systémově informační paket
F0–F7	240–247	Použití u prototypů

U světelných zařízení se typicky setkáváme s kanálem pro ovládání celkové intenzity, kanály pro samostatné ovládání složek RGB (případně jednoho kanálu na výběr výrobcem definovaného barevného mixu), kanálem pro ovládání tzv. GOBO (šablona, která slouží k promítání světelných tvarů), kanály pro ovládání horizontálního pohybu, vertikálního pohybu a jejich rychlosti (u otočných hlav) nebo kanálem

pro spuštění a ovládání rychlosti stroboskopu. Většina světelných zařízení pak umožňuje režim samostatného provozu, kde lze hodnoty nastavit ručně, přímo na zařízení (bez využití ovládání pomocí DMX512). V takovém případě lze využít zařízení jako vysílač signálu DMX512 a ovládat s ním ostatní zařízení připojená na jeho výstup, viz 1.2.1.

Z rozsahu 512 adres vyplývá zřejmá limitace protokolu na pouhých 512 samostatných komunikačních kanálů. V praxi se pro případy, kdy je potřeba tímto protokolem ovládat větší množství zařízení a parametrů, používají světelné pulty s několika DMX512 sběrnici [7].

1.2 Fyzická vrstva

Fyzická vrstva protokolu DMX512 vychází ze standardu RS-485, respektive EIA-485-A. Tento standard definuje napěťové úrovně a topologii spojení. Jeho základem je buď dvouvodičové, half-duplex nebo čtyřvodičové, full-duplex zapojení. V případě obousměrné half-duplex komunikace musí implementovaný protokol zajistit přepínání přijímače na režim vysílání a přijímání, dle jeho koncepce.

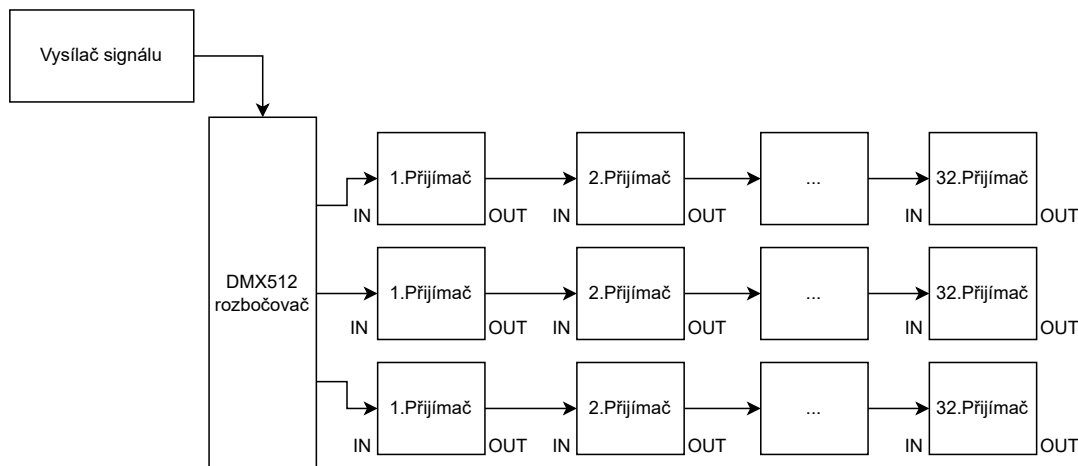
DMX512 používá pouze jednosměrnou komunikaci, a proto využívá dvouvodičové zapojení bez přepínání přijímač/zdroj signálu. Výhodou je jednoduchá a levná konstrukce přijímačů, nevýhodou potom nemožnost implementace obousměrné kontroly odeslaných a přijatých dat. Díky tomu není tento protokol určený k ovládání nebezpečných efektů [3] (např. pyrotechnických), které by při chybném spuštění mohly zranit účinkující.

DMX512 stejně jako RS-485 využívá primárně dvou datových vodičů a jednoho zemnicího vodiče. Ve standardu RS-485 se datové vodiče značí jako A, B a zem jako C, ve standardu DMX512 se používá označení Data 1–, Data 1+ a Data Link Common. Využívá se principu diferenciálního signálu – signál je veden vodičem A a B s navzájem opačnou polaritou. Výsledný signál je rozdíl mezi oběma vodiči. Výhodou je větší odolnost vůči rušení (případný rušivý signál se indukuje na oba dva vodiče zároveň, čímž se ve výsledku odečte).

Standard protokolu DMX512 obsahuje kromě primárních datových vodičů také dva sekundární datové vodiče (Data 2– a Data 2+), vedené stejným konektorem a kabelem. Jejich implementace do zařízení není povinná, využívají se totiž k přidání dodatečných funkcí k stávajícímu protokolu. Standard však povoluje použití jen definovaných funkcí, popsanych v příloze B standardu DMX512 [3] a v kapitole 1.2.5 (před zavedením jednoznačně definovaných funkcí druhého datového páru docházelo ze strany výrobců zařízení k porušování standardu EIA-485-A).

1.2.1 Topologie

Přijímače DMX512 mají zpravidla 2 konektory pro DMX signál – vstupní a výstupní. Vstupní konektor přivádí signál do zařízení a výstupní konektor odesílá nemodifikovaný vstupní signál dále. To umožňuje zapojovat přijímače do kaskády (viz blokové schéma 1.2). Omezení počtu zařízení je kvůli fyzické vrstvě RS-485 stanoveno na 32 přijímačů na jednu signálovou větev [3, 8], tedy maximálně 32 zařízení propojených za sebou. Proto se využívá aktivních rozbočovačů, které umožňují rozdělit signál do více větví a připojit tak více zařízení k jednomu vysílači.



Obr. 1.2: Příklad topologie sítě DMX512.

1.2.2 Používané konektory

Standard DMX512 předepisuje pro propojování zařízení konektory typu 5-pin XLR a IEC 60603-7 (označovaný jako RJ45).

Konektor 5-pin XLR je určen pro běžné použití u DMX512 zařízení. Zapojení se realizuje dle tabulky 1.3. Obvykle se na zařízení jako výstupní konektor používá samice, jako vstupní naopak samec.

Konektor IEC 60603-7 je určen pouze pro fixní instalace (např. uvnitř budov), které jsou za běžných podmínek přístupné pouze pověřeným osobám, tzn. není určený pro opakované připojování a odpojování zařízení. Zapojení se realizuje dle tabulky 1.4. Číslování pinů je odvozeno ze standardu ANSI/TIA/EIA-568 dle schématu T568B. Piny 3 a 8 jsou volitelné a nemusí být připojeny – záleží na použití. Pin 8 by však měl být zapojený nezávisle na absenci pinů 3 a 8, aby nedocházelo k rozdílnému potenciálu pinů 7 a 8.

Tab. 1.3: Pinout pro konektor 5-pin XLR [3].

Pin	DMX512 funkce
1	Data Link Common
2	Data 1–
3	Data 1+
4	Data 2–
5	Data 2+

Tab. 1.4: Pinout pro konektor IEC 60603-7 [3].

Pin	Barva	Funkce
1	oranžovo-bílá	Data 1+
2	oranžová	Data 1–
3	zeleno-bílá	Data 2+ (volitelný)
4	modrá	nepřiřazený
5	modro-bílá	nepřiřazený
6	zelená	Data 2– (volitelný)
7	hnědo-bílá	Data Link Common (Data 1)
8	hnědá	Data Link Common (Data 2)

Pro všeobecné použití tam, kde nemůže být z důvodu fyzické konstrukce zařízení použit 5-pin XLR konektor, je možné použít konektor alternativní, dle pravidel popsaných ve standardu DMX512 [3].

Kromě výše zmíněných konektorů se v praxi velmi často používá konektor 3-pin XLR, který není standardem povolený. Kabely s 3-pin XLR konektorem (samice–samec) se totiž často vyskytují jako kabely určené pro audiotechniku (zejména k připojení mikrofonů) a mají proto odlišné parametry – zejména impedanci a způsob stínění. Standard proto, kvůli možné záměně kabelu RS485/DMX512 s „mikrofonním“ kabelem, předepisuje pouze konektory XLR 5-pin, IEC 60603-7 a určité alternativní konektory. V praxi se pro 3-pin XLR používá pinout 1.5. Obvykle se na zařízení jako výstupní konektor používá samice, jako vstupní naopak samec.

1.2.3 Kabely a terminace signálu

Nároky na terminátor a impedanci propojovacích kabelů jsou rovněž stanoveny standardem RS-485. Impedance propojovacího kabelu je standardem RS-485 doporučena na $120\ \Omega$ [8] a jeho maximální délka není standardy definována, nicméně nedoporu-

Tab. 1.5: Pinout pro konektor 3-pin XLR.

Pin	DMX512 funkce
1	Data Link Common
2	Data 1–
3	Data 1+

Tab. 1.6: Tabulka doporučených řešení zemnění [4].

Skupina	Preferováno	Povoleno	Nepreferováno
Vysílač s referencí proti zemi	ANO		
Izolovaný vysílač		ANO	
Izolovaný přijímač	ANO		
Neizolovaný přijímač		ANO	
Přijímač spojený se zemí			ANO
Plovoucí přijímač/vysílač		ANO	

čuje se používat spojení delší než 300 m na jednu větev signálu [4, 9]. Toto délkové doporučení vychází z různorodosti přijímacích a vysílacích zařízení. V praxi se používají stíněné dvou nebo čtyřžilové, 110 Ω kabely [10] (viz 1.2.2).

Terminátor zabraňující odrazu signálu na vedení se připojuje mezi Data+ a Data– posledního řazeného přijímače v kaskádě. Terminátor by měl splňovat impedanci 120 Ω +5 %/–10 % [3]. Některá zařízení poskytují vnitřní terminaci datového signálu [3]. Komponenty terminátoru by měly být voleny tak, aby zvládly odolávat trvalému střídavému napětí 30 V 50 Hz nebo stejnosměrnému 42 V [3].

Konektory používaných kabelů jsou nejčastěji v konfiguraci samice–samec.

1.2.4 Elektrická konstrukce – izolace, zemnění zařízení

Standard DMX512 dělí elektrická řešení přijímačů a vysílačů do tří skupin, dle způsobu uzemnění zařízení a spojení země s Data Link Common. Jedná se o skupiny: preferovaných řešení – způsoby jakými by se zařízení měla stavět, povolených – způsoby jakými je zařízení také možno stavět a nepreferovaných řešení – způsoby, kterých je lepší se vyvarovat, nicméně jsou standardem povoleny [4]. Jednotlivá řešení jsou popsána v tabulce 1.6.

1.2.5 Přídavné funkce protokolu

Standard DMX512 rozděluje zařízení s přídavnými funkcemi do skupin (topologií) EF1, EF2, EF3 a EF4.

Skupina **EF1** se používá k označení přístrojů, které pomocí primárních datových vodičů využívají obousměrné komunikace half-duplex, splňující standard EIA-485-A. K funkci se používají alternativní Start kódy. Sekundární pár vodičů není využíván.

Skupina **EF2** se používá k označení přístrojů, které používají sekundární datové vodiče pro opačnou komunikaci (data z ovládaných zařízení směřují na vysílač). Jedná se tedy o obousměrnou komunikaci full-duplex, splňující EIA-485-A.

Skupina **EF3** vychází ze skupiny EF2, ale na sekundárním páru vodičů komunikuje obousměrně, half-duplex, v souladu se standardem EIA-485-A.

Skupina **EF4** pak využívá obousměrné komunikace na obou dvou párech, v souladu se standardem EIA-485-A.

Tyto topologie rozšiřují možnosti DMX512 například o zpětnou kontrolu odeslaného signálu, nebo informování přijímačů o jejich stavu přímo na ovládacím zařízení. Jednotlivé aplikace však záleží na výrobcích.

1.3 Komerčně dostupné analyzátory

Za účelem testování přijímačů, vysílačů i používané kabeláže je možné zakoupit některý z komerčně dostupných DMX512 analyzátorů/zkoušečů. Mezi jejich funkce patří zobrazení přijímaných dat na displeji, generování dat, analýza časování signálu, detekce a konfigurace Remote Device Management (RDM – rozšíření protokolu DMX512 o vzdálené nastavení zařízení), zobrazení napěťových úrovní signálu, nahrávání a přehrávání signálu, zkoušení kabelů. Zpravidla jsou provozovány na baterii a je možné je nabíjet. Přístroje jsou spíše menších rozměrů, určené na zkoušení (analýzu) v „terénu“.

Mezi zástupce starších přístrojů patří Artistic Licence Micro-Scope 3a, který se ovládá navigačními tlačítky a jako zobrazovací prvek používá displej s rozlišením 16x2 znaků [4]. Z novějších přístrojů je k dispozici Swisson XMT-500, který podporuje RDM a disponuje dotykovým displejem. Z levnějších zařízení je to pak Showtec DMX tester. Většina dostupných analyzátorů začíná s cenou v době psaní práce na přibližně 10000,– Kč, lepší analyzátory se pak pohybují zhruba na dvojnásobku zmíněné částky [11].

Cílem práce tedy bude vytvořit funkční analyzátor především s možností zobrazení signálu na PC a s možností generování. Zařízení však bude konstruované tak, aby šlo dále rozšiřovat o zmíněné funkcionality, které není možné v rámci této práce zcela obsáhnout.

2 Návrh a realizace prototypu

Cílem práce bylo navrhnout převodník USB–DMX512, který by spolu s počítačovým softwarem umožnil zobrazovat data zachycená na sběrnici DMX512, mohl je generovat, případně upravovat a mohl být použitý v laboratorní výuce. Mezi hardwarovými požadavky na zařízení byly informační a ovládací prvky, případně ukládání scén a nastavení na SD kartu a jejich opětovné načtení.

Při HW návrhu zařízení je možné bavit se minimálně o dvou možných řešeních. První řešení, které bylo do návrhu částečně implementováno je hardwarově velmi minimalistické. Jelikož je signál DMX512 asynchronní, sériový, generovaný UARTem a přenášený na fyzické vrstvě RS-485, je možné se k němu stavět stejně jako k ostatním UARTem generovaným signálům. Z napěťové vrstvy RS-485 jej lze převést na úroveň transistorové logiky (TTL) a následně například Serial–USB převodníkem odeslat do PC, kde se o analýzu signálu postará připravený software. Stejným způsobem pak lze realizovat i odesílání dat.

Toto řešení však není vhodné především kvůli počátečnímu požadavku na alespoň částečnou nezávislost analyzátoru na PC, kvůli požadavkům na displej, či SD kartu. Požadavek na případnou úpravu přicházejícího signálu by pak mohl být náročnější na praktickou realizaci z důvodu rychlosti přístupu operačního systému k virtuálnímu COM portu (mohlo by vznikat okem viditelné zpoždění na zařízeních zapojených v kaskádě za analyzátořem).

Druhé řešení opět využívá způsobu napěťových převodníků a rozhraní UART, obsahuje však navíc mikrokontrolér, který zde kromě ovládání uživatelských periférií zastává formu jakéhosi „rozbočovače“ analyzovaného signálu – přijímá analyzovaný signál a posílá jej na vstup PC a výstup analyzátoru. V případě režimu modifikace vstupního signálu pak ještě přijímá výstup PC, kombinuje jej se vstupním signálem a opět odesílá na výstup. Návrh je tak samozřejmě rozšířen i o program mikrokontroléru, který výše zmíněné funkce realizuje.

Tato bakalářská práce se bude věnovat právě řešení s mikrokontroléřem, které nabízí širší možnosti dalšího využití (přijímače, vysílače, dekodéry signálu pro LED, barevná hudba, apod.).

2.1 Výběř vhodného mikrokontroléřu

Při návrhu zařízení byl vybírán mikrokontrolér, který by umožňoval realizaci všech zmíněných požadavků a byl vhodný i pro případné budoucí rozšiřování analyzátoru o další funkce. Byl kladen důraz na HW výbavu mikrokontroléřu, jeho výkon a podporované instrukce, robustnost, trvanlivost, ale také dostupnost a cenu. V následujících sekcích budou krátce představeny dva nejvíce zvažované mikrokontroléřy

a provedeno jejich stručné srovnání a zdůvodnění volby.

2.1.1 ESP32

ESP32 je řada mikrokontrolérů vyvíjená společností Espressif Systems. Mikrokontroléry ESP32 se vyznačují především nízkou cenou, vysokým výkonem a integrovanou Wi-Fi, díky které se hodí na různé aplikace například v IoT (Internet of Things) [12]. Řada ESP32 obsahuje desítky 32-bitových modelů, které se liší především v použitém procesoru (počtem jader, takty), přítomností/absencí a verzí Bluetooth nebo množstvím vstupně/výstupních portů. Na trhu se objevují od roku 2016 [13].

Jako příklad rodiny ESP32 byl vybrán mikrokontrolér ESP32-S3-WROOM-1 se specifikacemi:

- CPU: Xtensa LX7, 32-bit, dvě jádra, až 240 MHz,
- ROM: 384 kB, SRAM: 512 kB, SRAM pro RTC: 16 kB, až 8 MB PSRAM,
- Wi-Fi: 2,4 GHz, 802.11 b/g/n,
- Bluetooth: verze 5,
- periférie: GPIO, 4x SPI, 2x I2C, 2x I2S, 3x UART, možnosti pro připojení LCD či kamery, a další,
- ADC: 2x 12-bit (až 18 měřicích kanálů),
- DAC: 2x 8-bit,
- pracovní teploty dle provedení od -40 až do 105 °C,
- cena: 69,12 Kč (k 29. 11. 2023, viz [15], nelze uplatnit množstevní sleva).

Jednotlivá provedení modelu se pak samozřejmě liší pouzdry, ale také například způsobem umístění antény pro bezdrátová připojení (běžné modely mají anténu integrovanou na desce s pouzdrům, modely s označením „U“ mají na pouzdře umístěný konektor pro externí anténu).

Pro programování mikrokontroléru se využívají různá vývojová prostředí, výrobcem podporovaná jsou Eclipse a Visual Studio Code. Podpora obou z nich spočívá v možnosti instalace rozšíření od Espressif Systems, které do prostředí zintegruje vlastní kompilátor i debugger a umožní tak psát a flashovat kód mikrokontroléru přímo z daného vývojového prostředí. Programování, respektive „flashování“ mikrokontroléru probíhá přímo, pomocí USB.

V rámci rozšíření je možné také vybrat jazyk, nebo lépe řečeno „framework“, ve kterém bude kód pro mikrokontrolér napsán. Kromě oficiálního frameworku ESP-IDF, který umožňuje vývoj v C/C++, a je tvůrci zamýšlený jako hlavní vývojový framework, je nabízena také možnost frameworku Arduino. Ten je velmi dobře známý především díky svojí jednoduchosti, respektive jednoduchosti kódu. Uživatelé zvyklí na prostředí Arduino z mikrokontrolérů ATmega, tak mohou snadno

přejít na výkonnější procesor vhodný pro značně pokročilejší aplikace (včetně analýzy zvukového či video signálu) a zároveň jej ovládat stejným jazykem, knihovnamí a funkcemi jako ostatní zařízení z rodiny Arduino.

V rámci frameworku ESP-IDF, je také možné používat tzv. FreeRTOS – operační systém, který umožňuje využívání tzv. „tasků“, podobně jako ve vyšších programovacích jazycích či operačních systémech [14]. To umožňuje snadnou práci s více jádry a multitasking.

2.1.2 STM32

STM32 je označení pro řadu mikrokontrolérů produkovaných firmou STMicroelectronics. Tyto mikrokontroléry jsou na trhu od roku 2007 a těší se velké oblibě zejména pro svoji stabilitu, podporu, dostupnost a cenu. Jednotlivých modelů z této řady je nepřehledné množství, liší se samozřejmě opět především v rychlosti procesoru, výbavě, ale také v možnostech integrovaného bezdrátového připojení. Žádný z modelů v době psaní práce nenabízí integrovanou Wi-Fi, výrobce se u modelů s bezdrátovým připojením drží spíše protokolů Bluetooth (až 5.4) a ZigBee. Oproti ESP32 však nabízí například integrované instrukce DSP (digitálního signálového procesoru), SAI (Serial Audio Interface) nebo oficiální knihovny a rozhraní pro tvorbu GUI (Graphical User Interface) [16], viz 4.3.

Jako příklad z rodiny STM32 byl pro srovnání parametrů vybrán mikrokontrolér STM32G431C6T6:

- CPU: Cortex-M4, 32-bit, jedno jádro s FPU, až 170 MHz,
- SRAM: 22 kB, FLASH: 128 kB,
- periférie: 86x GPIO, 3x SPI, 3x I2C, 2x I2S, 4x UART, možnosti pro připojení LCD či kamery, 1x SAI, UCPD a další,
- ADC: 2x až 16-bit (až 23 měřicích kanálů),
- DAC: 4x 12-bit,
- pracovní teploty dle provedení od -40 až do 105 °C,
- cena: 83,76 Kč (k 30. 11. 2023, viz [17], lze uplatnit množstevní sleva).

Pro programování mikrokontrolérů z řady STM32 lze využít oficiální vývojové prostředí STM32CubeIDE, které vychází z Eclipse a zastupuje původní Atollic TrueStudio. Prostředí podporuje jazyky C/C++ a obsahuje všechny potřebné komponenty pro vývoj na této řadě mikrokontrolérů, včetně možnosti debugování. Je také vzájemně kompatibilní s aplikací STM32CubeMX.

STM32CubeMX umožňuje konfigurovat periférie mikrokontroléru přehledně, pomocí grafického rozhraní. Aplikace nejprve vyzve k výběru uživatelem požadovaného modelu mikrokontroléru a následně zobrazí jeho pinout a umožní měnit nastavení

parametrů sběrnic, vstupně/výstupních pinů a dalších funkcí vybraného mikrokontroléru. Aplikace dále umožňuje nastavit funkce jednotlivých vývodů – například zda bude vývod fungovat v režimu GPIO-Input, GPIO-Output, GPIO-Analog nebo bude sloužit jako pin sběrnice. V případě, kdy funkce vývodu kolidují, což se stává především u sběrnic (vývod, který sběrnice využívá, je přidělen ještě jiné funkci), přehledně zobrazí v čem kolize spočívá, jaké sběrnice a jakým způsobem ovlivňuje a které sběrnice zůstávají nadále využitelné. Dále lze v programu nastavit funkce interních oscilátorů, časovačů ale také přerušování a DMA (Direct Memory Access). Při zakládání projektu je tak možné rychle zvolit přesná nastavení mikrokontroléru a program následně vygeneruje kód inicializující jednotlivé použité součásti. Použití aplikace však není nutné a inicializační kód si může programátor napsat sám dle potřeby.

Flashování mikrokontrolérů STM32 probíhá pomocí zařízení ST-Link. To se zapojuje do USB osobního počítače a pomocí SWD (Serial Wire Debug) se propojí s mikrokontrolérem. Nástroj následně umožňuje i debugovat chod programu. Vývojové desky STM32 často ST-Link integrují na DPS, a tím umožňují flashovat desku jako celek pomocí USB bez dalších převodníků [16].

K programování mikrokontrolérů STM32 lze použít tzv. HAL (Hardware Abstraction Layer) [18]. Jedná se o knihovny obsahující základní funkce „vyššího charakteru“, které zastupují práci s registry mikrokontroléru. HAL knihovny zároveň umožňují fungování jednoho kódu na vícero řadách STM32, jelikož názvy jednotlivých funkcí jsou unifikované napříč řadami a liší se pouze v jejich obsahu dle zařízení. V potaz se samozřejmě bere takový kód, který využívá pouze funkcí z knihoven HAL a nezasahuje do registrů mikrokontrolérů přímo.

Pro STM32 existuje také neoficiální Arduino framework [19], často se však dává přednost knihovnam HAL, které jsou oficiální, výrobcem podporované, zaměřené pouze na STM32. Podobně existuje i distribuce FreeRTOS pro STM32 [20].

2.1.3 Volba a porovnání

Rozhodnutí, který mikrokontrolér použít v rámci práce, bylo složité. Oba jmenované mají svá pro i proti. Navíc existuje velké množství alternativních nezmíněných mikrokontrolérů, které v práci mohly být použity. Do práce však byla vybrána rodina STM32 a tato podkapitola bude věnována konkrétním důvodům volby.

Prvotní myšlenkou bylo použít v práci moderní platformu, která je běžně aplikována v průmyslu a její znalost má tedy alespoň částečné uplatnění v praxi. I z těchto důvodů byl z výběru vypuštěn projekt Arduino a jeho SW implementace do moderních mikrokontrolérů. Výrobci Arduina jej považují spíše za prototypovací nástroj [21], což se neslučuje s výsledným použitím zařízení, kterému se věnuje tato práce.

Přístupování k HW mikrokontroléru pomocí Arduino frameworku navíc může způsobit pomalejší reakce HW [22], což by pro práci tohoto charakteru nemělo být nutně překážkou, ale ani potřebnou vlastností.

Před rozhodnutím, zda v práci použít ESP32 nebo STM32, byla zkušebně nainstalována obě oficiální programovací prostředí (Visual Studio Code s rozšířením a STM32CubeIDE) a na vývojových deskách byla otestována jednoduchá komunikace pomocí UART. Cílem bylo vyzkoušet přívětivost obou dvou prostředí, kvalitu oficiální dokumentace výrobců a obecně princip práce s mikrokontrolérem. Příprava obou prostředí byla bezproblémová u ESP32 však bylo hned po instalaci nutné upravit nastavení projektu pomocí komunitních návodů, jelikož prostředí zvýrazňovalo některý syntax mikrokontroléru, jako chybný.

Vyhledávání podkladů k práci s mikrokontrolérem bylo o poznání snažší u STM32. Kromě oficiálních manuálů a příruček bylo možné najít i velké množství komunitních materiálů přívětivých pro začátečníky, čímž se nejvíce lišilo oproti ESP32. Tam je dominantní především oficiální internetový manuál, zaměřený spíše na čtenáře více znalé problematiky mikrokontrolérů. Komunitní zdroje a fóra o ESP32 se velmi často zaměřují spíše na Arduino framework a zlehka opomínají možnou práci začátečníků s ESP-IDF. Větší četnost komunitních návodů k STM32 lze pravděpodobně přisuzovat především jeho stáří. Je zde také nutno podotknout, že kvalita dokumentace byla posuzována relativně, z pohledu „začátečníka“ v problematice.

Dalšími rozdíly byla doba nahrávání kódu (flashování) do mikrokontroléru. V defaultních konfiguracích bylo flashování STM32 znatelně rychlejší s rozdílem zhruba jedné minuty. S největší pravděpodobností by dobu flashování bylo u ESP32 možné snížit, nicméně je nutné podotknout, že tento fakt snižuje na počáteční přívětivosti a celkovém dojmu.

Velkou výhodou ESP32 byla samozřejmě Wi-Fi a Bluetooth spolu s dvoujádrovým procesorem. I přesto, že v zadání požadavek na Wi-Fi ani Bluetooth není, jednalo by se o zajímavé rozšíření práce. Takové řešení by šlo samozřejmě provést i na STM32 doplněním Wi-Fi integrovaného obvodu (např. ESP 8266), oproti ESP32 by však celková cena zařízení byla vyšší. I když je možné debatovat zda je Wi-Fi v laboratorním zařízení potřebná, užitek by mohla přinést v jiných aplikacích této práce.

Z HW výbavy se dále STM32 odlišuje bohatším portfoliem modelů. V závislosti na ceně se mění především HW – CPU, výbava, rozlišení převodníků, počet sběrnic, ... – což umožňuje uživateli vybrat zařízení v takové cenové hladině/s takovou výbavou, jaká je v projektu požadována. To má výhodu především při dodatečných změnách v projektu. V případě, že konstrukce probíhá například na zařízení z řady „mainstream“, je vždy možné se v případě např. nedostatku periférií přesunout na lépe vybavený model. U ESP32 však portfolio modelů tak bohaté není

a jejich výbava je napříč modely víceméně podobná.

HW výbava mikrokontroléru byla zvažována i v souvislosti s všestranností zařízení. V rámci práce má být vytvořeno zařízení především pro účely analýzy. V případě snahy o tvorbu koncepčně jiného zařízení pracujícího s DMX512 signálem, je možné z této práce použít většinu obsahu. Pokud by například byl podnět k vytvoření automatické barevné hudby, je možné využít HW/SW návrh této práce. STM32 by pak v takovém případě bylo vhodné k realizaci jak kvůli integrovaným DSP instrukcím, tak kvůli integraci SAI. Naopak v případě potřeby tvorby zařízení, které cílí na nízkou cenu (například ovladač osvětlení s pokročilými funkcemi), je STM32 vhodné kvůli širokému portfoliu modelů nabízejícímu i velmi levná řešení. Tento přístup s pohledem na „možné budoucí využití“ však není naprosto směřodatný. V případě, že by jedním z požadavků zařízení byla nízká cena a Wi-Fi, je např. ideální použít ESP32. Z toho obecně vyplývá, že volbu mikrokontroléru dle budoucího využití má smysl řešit pouze do určité míry, dle požadavků potenciálního zadavatele.

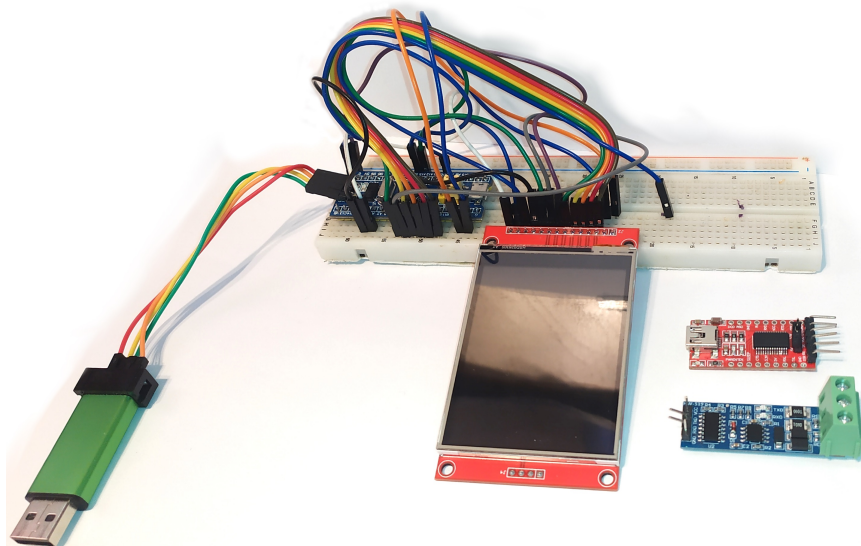
Právě z těchto důvodů byla pro realizaci vybrána rodina mikrokontrolérů STM32. V žádném případě to neznamená, že by STM32 byla oproti ESP32 lepší nebo naopak. Pro realizaci analyzátoru se STM32 pouze ukázalo jako stabilní a méně časově náročné řešení, což výběr zařízení ztlačilo. V rámci omezeného času na bakalářskou práci, kde není primárním záměrem sériová výroba jsou cena a rozdíly parametrů mikrokontrolérů zanedbatelné, ovšem komfort a časová náročnost při práci s nimi ne.

Pokud tedy práce místy působila jako „souboj dvou mikrokontrolérových rodin“, rád bych tento „boj“ ukončil. Obecně nelze určit jeden nejlepší mikrokontrolér. Pro každé řešení je vhodný jiný, a to ne pouze z jedné rodiny. Navíc nelze hledět pouze na technické parametry. Finanční stránka, koncept, čas na vývoj – to vše hraje při výběru velkou roli a teprve součet všech aspektů nabídne opravdu vhodný mikrokontrolér pro danou práci.

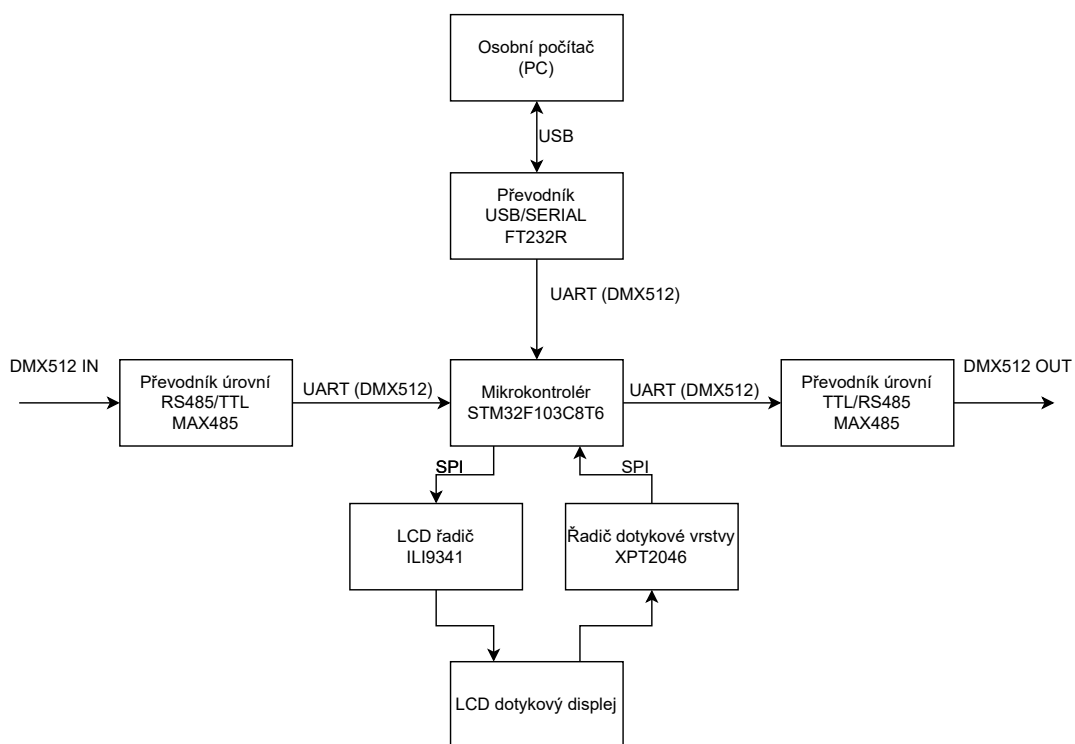
2.2 Realizace prototypu

V rámci semestrální práce byl navržen a zkonstruován prototyp analyzátoru, kterému bude věnována i tato podkapitola. Již v rámci semestrální práce bylo zvoleno řešení s mikrokontrolérem, právě kvůli jeho možnostem rozšíření o další užitečné funkce. Princip prototypu je zobrazen v blokovém schématu 2.2, v prototypu však nebylo realizováno čtení signálu.

Pro vývoj a realizaci prototypu byla použita neoficiální vývojová deska, komunitou nazývaná jako „Blue Pill“ [23]. Největší předností této desky je extrémně



Obr. 2.1: Prototyp analyzátoru.



Obr. 2.2: Blokové schéma prototypu.

nízká cena a zároveň dostatečná výbava. Vývojová deska je osazena mikrokontrolérem STM32F103C8T6 s jednojádrovým, 72 MHz procesorem, dostatkem sběrnic i vstupně/výstupních portů. Samotný procesor je výrobcem marketingově řazen

do skupiny „mainstream“. Desky však bývají osazeny mikrokontroléry z neoficiálních zdrojů (často se jedná o neoficiální klony) [24], což v praxi komplikuje flashování (STM32CubeIDE může hlásit chybu o neznámém mikrokontroléru) a teoreticky může způsobit problémy i při provozu mikrokontroléru. Při vývoji na Blue Pill je vhodné brát tuto informaci v potaz a případně alespoň jednu desku osadit mikrokontrolérem z oficiální distribuce, kvůli řešení možných chyb. Na druhou stranu s cenou 1,41 \$ za jednu desku (k 9. 12. 2023, [25]), je možné je používat jako velice levný vývojový prostředek k tvorbě většího množství prototypů.

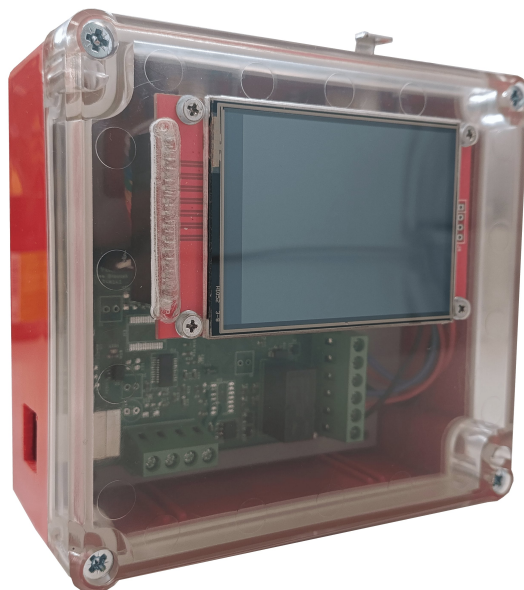
Prototyp pro komunikaci s PC využíval IO FTDI FT232R osazeném na komerčně dostupné desce s USB portem. Pro USB–Serial převod však byly testované i převodníky SiliconLabs CP2102 a WCH CH340G. Nejvíce se však osvědčil integrovaný obvod od FTDI, i kvůli svojí kompatibilitě s DMX512 řídicími aplikacemi, viz 3.2.3.

Pro převod napěťových úrovní z TTL na RS485 a naopak byly použity opět komerčně dostupné desky osazené pravděpodobně klonem integrovaného obvodu MAX485. V rámci prototypu se převodníky potýkaly s přehříváním, a proto byly do bakalářské práce obměněny, viz 3.2.2.

Napájení prototypu bylo řešeno pomocí ST-Linku, z USB portu.

3 Realizace analyzátoru

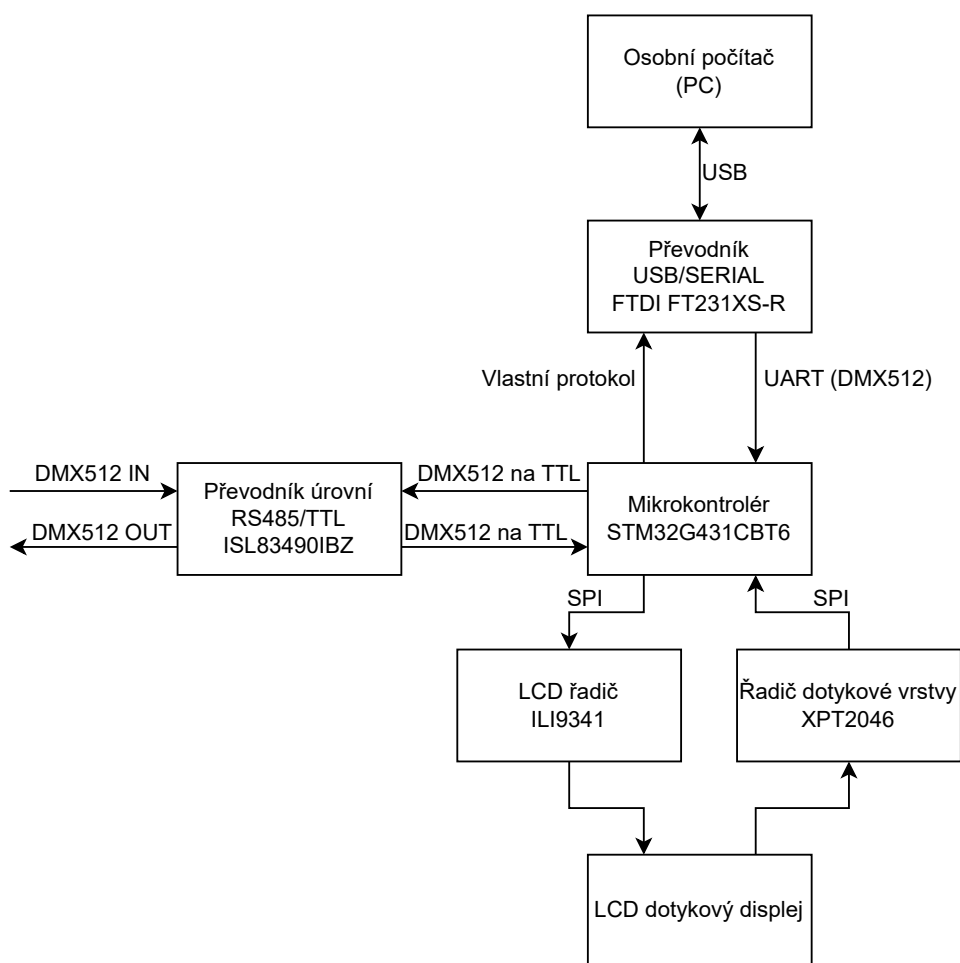
V rámci bakalářské práce byl vytvořen funkční přípravek pro analýzu a generování signálu DMX512. Ten využívá vlastního návrhu desky plošných spojů a parametry převyšuje původní prototyp. Tato kapitola se proto bude věnovat procesu jeho tvorby, návrhu a výběru součástek, tvorbě desky plošných spojů, popisu zařízení a jeho správné obsluze. V příloze C lze nalézt schéma konstruovaného analyzátoru, v příloze D potom nákres desky plošných spojů.



Obr. 3.1: Zkonstruovaný analyzátor v krabičce.

Zkonstruované zařízení se skládá z osazené desky plošných spojů, dotykového displeje a vstupních/výstupních konektorů. Specifikace výrobku jsou

- mikrokontrolér: STM32G431CBT6 – 170 MHz, 128 kB flash,
- displej: 2,8" 320x240 TFT, barevný, dotykový,
- provozní napětí: 1,4 V–20 V nebo 5 V z USB,
- konektor pro PC: USB typ B,
- konektory pro DMX512: XLR 3-pin,
- rozměry: 125,3 mm x 110 mm x 52 mm,
- hmotnost: 393 g,
- kompatibilita s běžnými ovládacími SW.



Obr. 3.2: Blokové schéma analyzátoru.

3.1 Funkce analyzátoru a jejich řešení

3.1.1 Analýza signálu

Hlavní funkcí zařízení je analýza signálu DMX512. Zpracování signálu v rámci HW probíhá dle blokového schématu 2.2. Signál je nejprve z diferenciálního páru vodičů fyzické vrstvy RS485, převeden na signál se společnou zemí a napěťovou úrovní TTL, kterou lze zpracovat mikrokontrolérem. UART mikrokontroléru přijímá data stále ve formátu DMX512 a jeho program je dále zpracovává, viz 5.4.1. Po uložení dat do paměti zařízení, mohou být zobrazeny přímo na displeji zařízení, nebo odeslány do počítače. Pro komunikaci s počítačem byl vytvořen vlastní protokol popsáný v podkapitole 3.1.3.

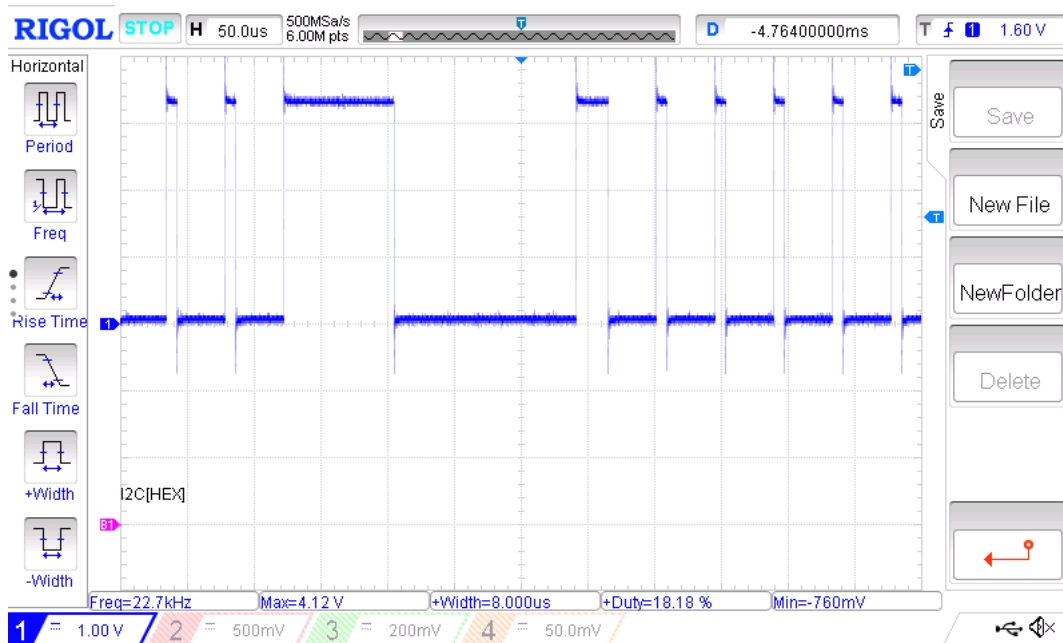
V režimu analýzy je analyzátor v síti zapojen jako přijímač. Ke sběrnici je tedy připojen paralelně. Vstup a výstup analyzátoru je vodivě propojen. Signál DMX512

je zachytáván v MCU, výstup MCU je zcela odpojen. V tomto režimu je zařízení možné zapojit doprostřed, nebo na konec sítě DMX512 tak, jako běžně dostupné přijímače (světla). Zařízení je tedy ke sběrnici DMX512 připojeno paralelně. Při použití výstupního konektoru jako pass-through nedochází k žádné modifikaci signálu.

3.1.2 Generování

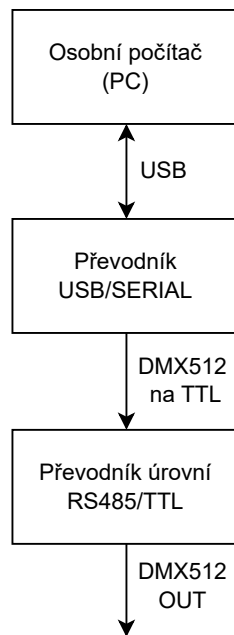
Generování využívá podobného principu. MCU tvoří pakety signálu DMX512, které vysílá na vstup TTL–RS485 převodníku, viz 4.2. V případě, že je ke generování signálu použita aplikace pro PC, přijímá MCU signál z PC ve formátu DMX512, zpracovává jej stejným způsobem jako signál z převodníku TTL–RS485 a následně ho odesílá jako DMX512 signál na svůj výstup.

V režimu generování je analyzátor v síti DMX512 zapojen jako vysílač. Vstup a výstup zařízení již není přímo vodivě propojen. Vstupní signál je z konektoru veden do MCU, výstupní signál je veden z MCU do konektoru. Zařízení tedy může být zapojeno jediné na začátku sítě, v opačném případě přerušuje cestu signálu.



Obr. 3.3: Analyzátořem generovaný signál – začátek paketu.

Toto řešení s přepínačem bylo implementováno kvůli zvažované možnosti editace signálu. V tom případě by bylo zařízení na sběrnici DMX512 zapojeno v sérii, v rámci sítě jako přijímač a zároveň vysílač. Vstup a výstup zařízení by nebyl vodivě propojen. V topologii DMX512 však v tomto případě nemůžeme mluvit o klasickém přijímači s pass-through výstupem. Formálně analyzátor jednu síť na svém vstupu



Obr. 3.4: Samostatné použití SW pro generování zpráv.

ukončí a vytvoří novou na svém výstupu, čímž se zachová jako zdroj signálu/opakovač a do jedné větve za něj lze tedy opět zapojit dalších 32 přijímačů. Výstupní signál je navíc vlivem časového zpoždění a variabilitou časování DMX512 takřka vždy technicky odlišný od vstupního, byť třeba přenáší stejné datové hodnoty.

Přepínání (spínání) vstupu a výstupu je v práci řešeno pomocí relé, viz podkapitolu 3.2.6.

Mimo zmíněné režimy lze vyvinutou aplikaci (viz kapitolu 5) používat i bez hardwaru analyzátoru, takřka s libovolným USB–Serial převodníkem a libovolným převodníkem napěťových úrovní na RS485, viz blokové schéma 3.4. K tomu je využito odesílání dat ze softwaru ve tvaru DMX512.

3.1.3 Vlastní protokol pro komunikaci

V rámci semestrální práce bylo zamýšleno použít k odesílání dat z analyzátoru do PC protokol DMX512. Formátování paketů a jejich časování by tak bylo shodné pro všechny druhy komunikace. Vzhledem k tomu, že je však formátování paketů v protokolu založeno především na časování jednotlivých prvků paketu, s čímž by při zpracování vyšším programovacím jazykem mohl být problém, byla zvolena tvorba vlastního protokolu.

Protokol využívá stejného nastavení rozhraní UART jako protokol DMX512, tedy 250 kbit/s baudrate, 1 start bit, 8 datových bitů (LSB první) a 2 stop bity.

Tab. 3.1: Formát vlastního protokolu pro odesílání.

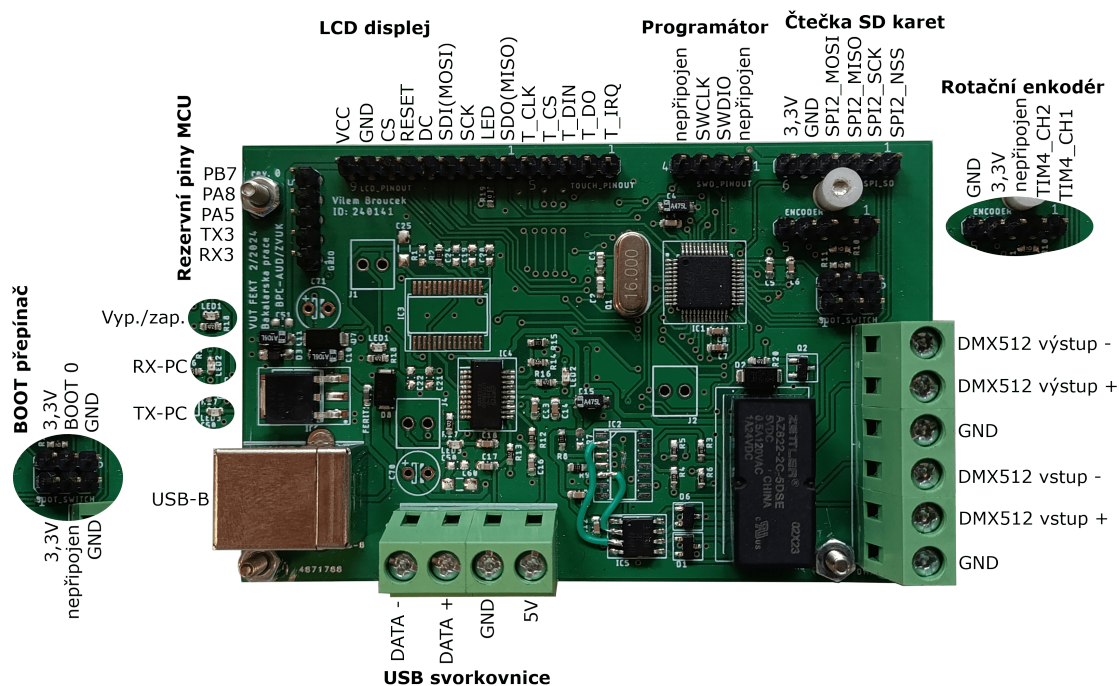
Pořadí bytu	Hodnoty bytu	Význam bytu
0	121	První část hlavičky
1	122	Druhá část hlavičky
2	přenášená data	Start code DMX512
3	přenášená data	1. datový byte DMX512
4	přenášená data	2. datový byte DMX512
...
512	přenášená data	510. datový byte DMX512
513	přenášená data	511. datový byte DMX512
514	přenášená data	512. datový byte DMX512
515–517	0	Rezervováno pro úpravy protokolu
518	131	První část patičky
519	132	Druhá část patičky

Jeden paket protokolu se skládá z 520 bytů (z toho 513 datových) a určení začátku a konce paketu se provádí dle hodnot přijatých v paketu – hlavičky a patičky. Hlavička paketu je vždy uložena v prvních dvou bytech paketu, patička ve dvou posledních, viz tabulku 3.1. Hodnoty hlavičky a patičky jsou fixní a v průběhu odesílání se nemění, lze je měnit pouze ve zdrojovém kódu. Mezi hlavičkou a patičkou se přenáší 513 datových bytů pro přenos signálu DMX512, a také 3 volné byty, které jsou rezervované pro další úpravy protokolu. Tyto byty by mohly obsahovat hodnotu kontrolní sumy (checksum) pro zjištění celistvosti a neporušenosti odesílaných paketů. Případně v nich lze posílat časový údaj – tzn. kdy byl paket přijat, nebo pomocí nich z počítačové aplikace ovládat nastavení HW analyzátoru. V rámci této práce jsou však nevyužité a odesílají se s nulovými hodnotami. V případě jejich nedostatku je lze jednoduše rozšířit pomocí konstantních proměnných v kódu, viz podkapitulu 5.4.1.

Systém hlavičky a patičky byl vybrán kvůli jednoduchosti komunikace. Vzhledem k algoritmu hledání začátků a konců paketu, může být nevýhodou teoretická chybovost. V případě že program nalezne datové byty (z komunikace DMX512), které mají stejnou hodnotu jako hlavička protokolu, a k nim navíc shodou náhod nalezne s odstupem 518 bytů dva byty odpovídající patičce, načte do paměti chybné hodnoty a s jejich výpisem bude pokračovat do doby, dokud se data přijímaná na sběrnici DMX512 v těchto konkrétních kanálech nezmění.

Této chybě se práce snaží předejít použitím dvou-bytové hlavičky a patičky s různými hodnotami a použitím rozpoznávacího algoritmu, který si po prvním paketu komunikaci zarovná na nultý byte a následující datové byty tak již nebere v potaz.

3.2 Popis hardware analyzátoru



Obr. 3.5: Osazená DPS analyzátoru s popisky konektorů.

3.2.1 Mikrokontrolér

Tato práce využívá mikrokontroléru STM32G431CBT6. Typově se jedná o MCU popisované v kapitole 2.1.2 pouze s rozšířením vnitřní flash paměti na 128 kB. Ta se totiž po osazení na desku u modelu STM32G431C6T6 ukázala být nedostatečná. Kód prototypu, provozovaný původně na STM32F103C8T6 s 32 kB flash paměti, se na zmíněný model, taktéž s 32 kB, nevmístit. Na vině při tom byly rozsáhlejší knihovny HAL, které u lepšího modelu zahrnovaly obsluhu více periférií a tím pádem byly objemově větší. Řešením by bylo nepotřebné části kódu knihovny odebrat, byla však zvolena výměna mikrokontroléru za model s větší pamětí flash, a to především kvůli celkové nejistotě nad finální velikostí programu. Tím byla vytvořena dostatečná rezerva flash paměti pro vývoj, která by s nyní známou velikostí kódu mohla být pro další verze desky spolu s cenou mikrokontroléru zmenšena.

MCU STM32G431CBT6 bylo vybráno pomocí interaktivního katalogu v aplikaci STM32CubeMX, který na základě pouzdra a konfigurace vstupně-výstupních pinů v *.ioc* souboru vybere pro stejný projekt kompatibilní mikrokontroléry z řady STM32. Kromě totožné řady bylo možné použít i některé vyšší řady, které mají stejné rozvržení periférií na pinech.

Pro mikrokontrolér byl vytvořen nový konfigurační soubor v CubeMX a překopírován zdrojový kód ze souborů prototypu. Zároveň byly upraveny vstupní a výstupní piny a konfigurace knihovny displeje.

Jak vyplývá z blokového schématu 3.2, mikrokontrolér zde slouží k přijímání, zpracování a odesílání DMX512 signálu, k obsluze dotykového displeje a vykreslování GUI.

3.2.2 Převodník RS485–TTL

Pro převod DMX512 signálu mezi napěťovými vrstvami RS485–TTL je použit integrovaný obvod ISL83490IBZ. Jedná se o full-duplex převodník s přenosovou rychlostí 10 Mbit/s (dostatečnou pro DMX512), který byl vybrán na základě nízké ceny a jednoduchého zapojení. Tento převodník nemá piny ovládající aktivitu přijímání a vysílání, na jeho pin tedy stačí přivést vstupní signál nebo odebrat signál výstupní. V rámci práce je díky vlastnosti full-duplex použit pouze jeden převodník pro vstupní i výstupní komunikaci na sběrnici DMX512.

3.2.3 Převodník USB–Serial

Pro odesílání signálu do počítače byl vybrán integrovaný obvod FTDI FT231XS-R. Jedná se o novější a levnější verzi v prototypu používaného obvodu FT232R, která si však zachovává veškerou pro práci podstatnou funkcionalitu, stejně jako shodné ovladače pro Windows.

Právě shodnost ovladačů hrála největší roli při výběru obvodu. Původní integrovaný obvod FT232R je totiž velmi často používaný v komerčně dostupných převodnících, např. Enttec Open DMX USB [26], což se ukazuje jako největší výhoda oproti ostatním v prototypu testovaným převodníkům. Většina specializovaného SW pro ovládání scénické techniky (např. FreeStyler [27]) totiž umožňuje propojení pouze s omezeným katalogem softwarem podporovaných USB–DMX512 převodníků. Enttec Open DMX USB je však díky vysoké popularitě jedním z velmi často podporovaných převodníků v polo-profesionálních i profesionálních řídicích softwarech a s ním samozřejmě i USB–Serial převodník FT232R a jeho příbuzné modely.

Použití integrovaného obvodu FT231XS-R tedy umožňuje analyzátoru generovat zprávy ne jenom pomocí vlastního SW, ale i ostatními běžně používanými SW pro ovládání scénické techniky.

Další výhodou integrovaného obvodu této řady je výrobcem nabízená možnost změny PID (ID produktu) a VID (ID výrobce) ovladače, což umožňuje přidělit analyzátoru unikátní název, který se bude zobrazovat v operačním systému. V defaultním

nastavení se totiž převodníky zpravidla zobrazují jako virtuální COM port s označením např. „COM 1“, kde se číslo v názvu mění dle ostatních zařízení připojených k PC (dle OS).

3.2.4 Napájení, napěťový regulátor LDO, spotřeba

Všechny vybrané komponenty využívají napěťovou úroveň 3,3 V. Pro regulaci napětí z USB, případně z napájecích svorek na 3,3 V je použit LDO (Low-dropout) regulátor ONSEMI NCV1117DT33T5G. S výstupním proudem až 1,5 A poskytuje skoro až přílišnou rezervu pro další rozšiřování funkcí desky. Zároveň nabízí dostatečný rozsah vstupních napětí pro případnou změnu napájení.

USB měřicím přístrojem Atorch U96P byla orientačně změřena i spotřeba zařízení při analyzování signálu v PC, která vychází zhruba na 384 mW (napájeno z USB, 5,12 V). V některých režimech může být spotřeba vyšší.

3.2.5 LCD displej

Jako hlavní ovládací prvek zařízení byl použit dotykový displej. Jeho cena je poměrně nízká, zároveň však činí práci se zařízením snažší, rychlejší a přehlednější. Rozlišení 320x240 na 2,8" velkém displeji je dostatečné pro vykreslení většího množství hodnot signálu a pro komfortní navigaci v menu. Dotyková vrstva displeje je odporová, k zařízení je proto přiložen i plastový stylus, viz 4.3.

Displej je součástí komerčně dostupné desky, která je osazena řadičem ILI9341, k ovládání vykreslování na displeji, a řadičem XPT2046, k ovládání dotykové vrstvy. Oba řadiče komunikují s mikrokontrolérem pomocí sběrnice SPI. Jas displeje je možné ovládat pomocí PWM (Pulse Width Modulation), v této iteraci desky je však připojen přímo na 3,3 V větev, pouze přes nulový rezistor (kvůli případnému fixnímu snížení jasu).

3.2.6 Relé

V zařízení je k přepínání vstupních a výstupních konektorů pro signál DMX512 použito relé AZ822-2C-5DSE s přepínáním DPDT (Double Pole, Double Throw) – dva přepínací kontakty (viz schéma C). V nepřepnutém stavu relé jsou spolu propojeny signálové vstupní a výstupní svorky DMX512. Analyzátor tak funguje pouze jako přijímač. Při aktivaci relé se vstupní a výstupní svorky rozpojí, vstupní zůstávají připojeny na vstup MCU, výstupní se propojí s výstupem MCU. Zem je pro oba stavy, vstupní i výstupní svorky, společná. Tato funkcionalita byla blíže popsána v předchozích kapitolách.

3.2.7 Externí konektory

Zařízení je vybaveno zmíněným konektorem USB-B pro připojení k počítači. Tento konektor byl vybrán kvůli jeho robustnosti. Pokud by bylo nutné desku zmenšit, volil by se konektor menších rozměrů, viz podkapitolu 3.5. V tomto případě však konektor i esteticky odpovídá celkovým rozměrům zařízení. Konektor je umístěn přímo na DPS, kvůli snažšímu debugování desky i horší dostupnosti panelových USB konektorů.

Použité panelové konektory 3-pin XLR samec (pro vstup signálu) a samice (pro výstup signálu) sice nejsou normou ANSI podporovány, i tak jsou však hojně používány především u levnějších světelných zařízení a DMX512 pultů. V případě potřeby analýzy/generování signálu do 5-pinového kabelu je možné použít komerčně dostupné redukce, případně takovou redukci propojením příslušných pinů vyrobit, viz 1.2.2.

Panelové konektory byly upřednostněny před konektory do DPS opět kvůli jednoduššímu debugování analyzátoru při jeho vývoji. Zároveň jsou robustnější (nelze je vylomit z DPS) a dají se případně snadno zaměnit za jiné, normou podporované, viz podkapitolu 1.2.2.

3.2.8 Interní konektory

Deska je osazena velkým množstvím konektorů, které jsou blíže popsány na obrázku 3.5. Hlavní zastoupení zde mají šroubovací svorkovnice s roztečí 5 mm. Určené pro připojení panelových konektorů XLR, případně pro vyvedení USB do panelu. Šroubovací svorkovnice byly zvoleny kvůli nízké ceně a velké užitečnosti při vývoji SW zařízení, kdy je vhodné mít do jedné svorky připojených několik dalších zařízení (zejména osciloskop, USB–Serial převodník a DMX512 přijímač – světlo). Tato testovací zapojení by v případě osazení XLR konektorů do DPS vyžadovala výrobu dalších přípravků a redukcí. Výhodou je také snadná výměna konektorů za jiné.

Menší šroubovací svorkovnice 2,54 mm jsou osazené pouze na jedné ze dvou vytvářených desek a jsou určeny pro připojení informačních LED diod – zařízení zapnuto, Tx aktivita USB–Serial převodníku, Rx aktivita USB–Serial převodníku.

Dále desky obsahují hřebínkové konektory s roztečí 2,54 mm, které nejsou pro zařízení příliš vhodné, ale v rámci první revize desky posloužily opět k ladění zařízení bez potřeby krimpovat na kabely specifické konektory. Na tyto konektory se připojuje například displej, programátor ST-Link, vyvedeno je připojení SD karty, rotačního enkodéru, přepínání boot režimu. Jedna z patič je na desce navíc, jsou na ní vyvedené některé nadbytečné piny MCU, včetně třetího UARTu, pinů podporujících A/D, D/A převod a podobně, pro řešení problémů při ožívování desky,

případně pro testování nových funkcí do další verze desky. Více k tomuto řešení v podkapitole 3.5.

Hřebínek na přepínání boot režimu lze osadit jumpery pro výběr režimu bootování (výběru flashování, apod.) dle schématu a referenčního manuálu MCU. Byl přidán na základě zkušeností s vývojovou deskou BluePill, v rámci práce však není využíván.

Pozor! V případě použití interní svorkovnice pro napájení, nesmí být využíván osazený USB-B konektor.

3.2.9 Informační LED diody

Deska namontovaná ve finálním výrobku je namísto 2,54 mm svorkovnicemi osazena třemi žlutými LED diodami popsány na obrázku 3.5. Jedna z LED diod je připojena přímo k 5 V větvi a signalizuje, zda je zařízení zapnuté. Zbylé dvě LED diody jsou ovládány obvodem FT231XS-R, jedna signalizuje aktivitu přijímání dat z PC, druhá odesílání do PC. Pozice LED diod byla volena takřka náhodně, neboť v původním plánu bylo vyvést LED diody na přední panel pomocí šroubovacích svorkovnic. Vzhledem k průhlednému víku, však byla vybrána varianta s osazenými LED.

3.2.10 Krabíčka

Zařízení je zapouzdřeno v červené krabíčce Z59PH s průhledným víkem, vyrobené z ABS plastu. Pestřejší barva krabíčky byla zvolena cíleně kvůli povaze zařízení – protokol DMX512 se používá k ovládání velkých světelných show, ani krabíčka by proto nemusela být vysloveně „smutná“. Díky průhlednému víku je DPS spolu s konektory a kabeláží viditelná. DPS je přišroubovaná na vlastní, frézované montážní desce, která je přišroubovaná do montážních úchytů krabíčky.

Displej je situován horizontálně, aby bylo na displeji zobrazitelné větší množství analyzovaných hodnot. Je umístěn do vrchní části víka tak, aby mohl být k zařízení v budoucnu přidán rotační enkodér, případně tlačítka, viz podkapitolu 3.5.

3.3 Konstrukce desky plošných spojů

Deska plošných spojů byla spolu se schématem navrhována v programu EAGLE a je vložena v příloze D. Při návrhu a výběru součástek se dbalo především na dobrou dostupnost použitých součástek a volbu pokud možno standardních pouzder. V rámci práce bylo do výroby zadáno 5 DPS, z čehož dvě byly (s drobnými vzájemnými rozdíly) osazeny. DPS jsou dvouvrstvé s nepájivou maskou a potiskem. U většiny

součástek byla, kvůli snažší montáži, preferována varianta SMD (Surface Mount Device), všechny konektory a relé jsou naopak v provedení THT (Through-hole technology) kvůli větší robustnosti.

Osazené rezistory a keramické kondenzátory používají jednotně pouzdro 0603. Integrované obvody na desce jsou napájeny jednotně 3,3 V.

Na desce jsou kromě zmíněných osazovaných IO nechány také volné plošky, které umožňují osazení odlišného integrovaného obvodu. Místo FT231XS-R tak lze osadit IO FT232RNL, místo převodníku ISL83490IBZ potom SN65HVD76DR, který zároveň disponuje více funkcemi. K tomuto řešení bylo přistoupeno, aby se dalo předejít možným komplikacím v případě, že zamýšlené IO nebudou fungovat podle představ. Konkrétně u FT231XS-R spočívala nejistota v kompatibilitě s ovládacími SW. V případě, že by zamýšlené IO nevyhovovaly, bylo by je možné nahradit bez výroby nové DPS. Toto „zdvojení“ některých součástek desce sice navyšuje rozměry, v rámci její první verze však na tento fakt nebyl brán zřetel. V případě osazení jiného IO je také potřeba osadit součástky k němu dle schématu náležící (schéma je rozděleno do celků).

Na desce jsou za účely ladění nechána volná místa na 5 V větvi pro keramický, tantalový a elektrolytický kondenzátor (C50, C60, C70). Na 3,3 V větvi je to pak pouze keramický a elektrolytický (C51, C71), tantalový je osazen (v případě problému je možné osadit místo něj jiný, s vyšší kapacitou).

SMD LED diody na desce jsou předřadnými odpory 4,7 k Ω regulovány na přijatelný jas. LED dioda pro signalizaci vypnutého/zapnutého stavu je zapojena přímo na 5 V. Zbylé LED diody jsou napájeny z 3,3 V větve, jejich svítivost je tedy menší. Jedná se o cílený krok, jelikož zbylé signalizační diody rychle blikají s příchozí/odchozí komunikací s PC, což by mohlo být při vyšším jasu rušivé.

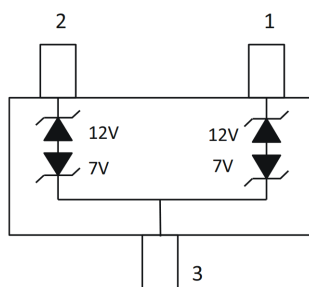
Relé osazené na desce není spínáno (respektive přepínáno) přímo vývodem mikrokontroléru, ale pomocí tranzistoru. K tomuto kroku bylo přistoupeno především kvůli omezené proudové zátěži vývodů MCU, kterou výrobce udává až na 20 mA na pin [28]. V rámci datasheetu relé je však udávána spotřeba vyšší – 98 mW, tedy přibližně 30 mA. V zapojení je tedy použit unipolární tranzistor s indukovaným kanálem typu N, BSS123, ovládaný vývodem MCU (s pull-down R20), který spíná 3,3 V napěťovou větev přiváděnou na vývody relé.

Jako ochrana proti přepětí jsou na desce použité transily. Transil D4 se závěrným napětím 5 V je použit na ochranu 5 V větve a je umístěn v závěrném směru za feritovou perlou, která potlačuje vysokofrekvenční šum. Podobně je tomu u D4, pouze na 3,3 V větvi. Schottkyho dioda SK15, D5, zabraňuje stavu, kdy se 5 V větev vybijí rychleji než 3,3 V. Stejná dioda D6 pak plní zhášecí funkci pro přepínací cívku relé.

Ochrana proti přepětí DMX512 zařízení a převodníku TTL–RS485 je řešena

pomocí transilů integrovaných v obvodu SM712 (doporučených v datasheetu převodníku TI). Na desce jsou osazeny dva obvody, pro vstup a výstup z převodníku. Transily v pouzdru jsou zapojeny v závěrném směru pro danou polaritu signálu, a mají takové závěrné napětí, jaké je maximální hodnota dané polarity pro vrstvu RS485 (tedy +12 V a -7 V), viz schéma 3.6.

Dle standardu DMX512 je přijímací strana zkonstruovaného analyzátoru kategorizována jako vysílač s referencí proti zemi (ground referenced), vysílací strana potom jako přijímač spojený se zemí (grounded).



Obr. 3.6: Transily na ochranu proti přepětí [29].

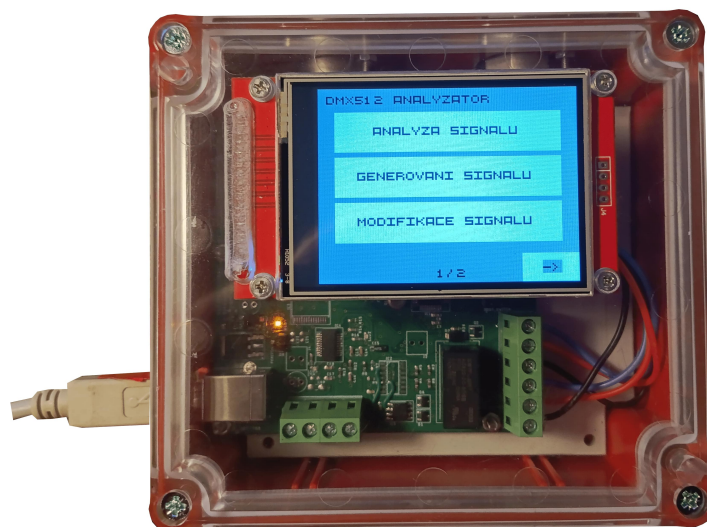
Po osazení a oživení desky byl zjištěn nevyhovující MCU, který musel být následně vyměněn za jiný, viz 3.2.1. Při oživování bylo zjištěno také prohození pinu Rx a Tx z MCU do převodníků TTL–RS485, které bylo vyřešeno přerušením cesty na DPS a následným propojením z volných plošek po neosazeném převodníku s vývody IO vodičem.

Pozor! V rámci schématu a exportů DPS vložených v této práci (C, D) jsou zmíněné chyby, spolu s několika dalšími (chybějící napájecí cesty do konektorů pro programátor, tloušťky cest pro boot patiči, ...), již opraveny – proto deska *rev. 0.1*. Neupravená verze *rev. 0*, korespondující s vyrobenou deskou, je k dispozici v elektronické příloze.

3.4 Obsluha zařízení

Pro spuštění zařízení je třeba jej nejprve připojit do USB. USB-B konektor je vyveden na levé boční stěně přístroje. Po zapojení se okamžitě rozsvítí LCD displej a signalizační LED dioda na desce. Na přední stěně přístroje jsou vyvedeny XLR konektory pro připojení DMX512, viz 3.9.

Na displeji se vykreslí menu se třemi tlačítky – volitelnými režimy: *Analyza signalu*, *Generovani signalu*, *Modifikace signalu*. V rámci menu je možné postoupit



Obr. 3.7: Hlavní menu analyzátoru.

na další stránku, kde jsou nabídnuty možnosti *Manuální analýza* a *Manuální generování*. V případě, že libovolná stránka obsahuje více „podstránek“, je ve spodní části zobrazena aktuální stránka, počet následujících stránek (1/2) a příslušná navigační tlačítka (←, →) pro navigaci mezi nimi.

Dotykem lze zvolit režim, ve kterém bude analyzátor pracovat. Každý režim využívá definovaných funkcionalit dle tabulky 3.2. Po zvolení se analyzátor přepne a zahájí patřičné funkce. Při aktivaci režimů využívajících PC, je na displeji zobrazeno pouze oznámení, aby se uživatel přesunul do počítačové aplikace. Při zvolení manuální analýzy se zobrazí matice čísel, které odpovídají číslu příslušného kanálu DMX512 a jeho hodnotě (ve formátu „číslo kanálu: hodnota“). V rámci této podstránky lze pomocí navigačních šipek zobrazit všech 512 kanálů přijímaného signálu.

MANUALNI ANALYZA		
1: 233	2: 214	3: 166
4: 170	5: 181	6: 68
7: 255	8: 8	9: 110
10: 133	11: 178	12: 0
13: 0	14: 0	15: 0
16: 0	17: 0	18: 0
	0: 0	
←	1/29	→

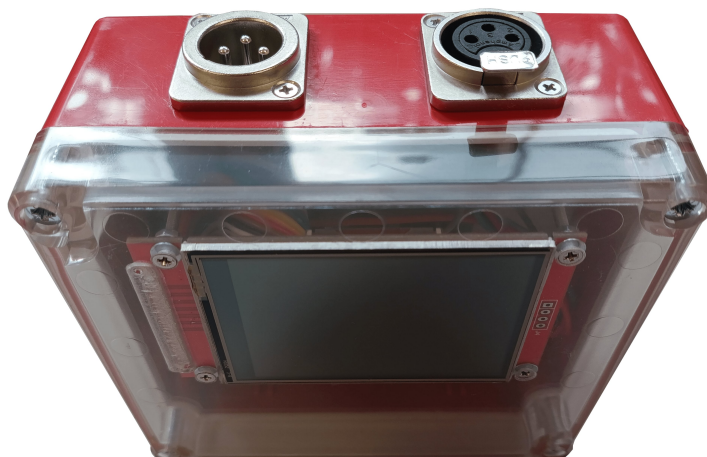
Obr. 3.8: Režim manuální analýzy.

Manuální generování slouží pouze pro demonstrační či testovací účely, neboť je naprogramováno pouze pro konkrétní model přijímače (světla). Obsahuje tři tla-

Tab. 3.2: Funkce analyzátoru dle zvoleného režimu.

–	Relé	Odesílání	Přijímání
Analýza	nepřepnuto	do PC	z DMX512
Generování	přepnuto	do DMX512	z PC
Modifikace	přepnuto	do PC, do DMX512	z DMX512, z PC
Manuální analýza	nepřepnuto	neodesílá	z DMX512
Manuální generování	přepnuto	do DMX512	nepřijímá

čítka – červené, zelené a modré, které nastaví příslušné barvy na světle testovaném v této práci. Světlo musí mít nastavenou adresu číslo 1 (mění se první 4 byty signálu, pro použité světlo jsou to 1. celková intenzita, 2. červená, 3. zelená, 4. modrá). Hodnoty jsou zvolené fixně v kódu, v rámci této práce je nelze jednoduše uživatelsky měnit.



Obr. 3.9: XLR konektory analyzátoru, vlevo vstup, vpravo výstup.

Návratem do hlavního menu (←) se zruší předchozí zvolený režim a zařízení ukončí odesílání/přijímání.

Obsluha aplikace pro osobní počítač je popsána v podkapitole 5.3. Modifikace singálu není v této verzi zařízení naprogramována.

3.5 Doporučené úpravy pro další verze

V rámci této práce bylo uskutečněno několik rozhodnutí, která se v průběhu, nebo až s hotovou realizací ukázala být méně vhodná. Neovlivňuje to sice přímo chod analyzátoru, jedná se ale o věci, které by bylo vhodné s příštími iteracemi desek (či verzí

kódu) zlepšit. Opravám chyb a návrhům na zlepšení zařízení se proto bude věnovat tato podkapitola.

Vzhledem k tomu, že se jedná o nultou revizi DPS, byla dle toho koncipována většina jejích vlastností. Jejím hlavním cílem bylo usnadnit ožívování, programování a celkovou realizaci zařízení. Na desce se proto nachází zmíněné nepoužité piny, nebo neosazené plošky IO a kondenzátorů, které ale v odladěné verzi desky již nemají takřka žádný význam. V rámci příštích revizí DPS, by tedy bylo vhodné celkově desku zmenšit, ořezat o nevyužívané funkce a rozměry koncipovat přímo do existujících, komerčně dostupných krabiček. Zařízení tak bude moct být menší a lépe tak bude konkurovat komerčně dostupným analyzátorům.

Pro zmenšení zařízení je také vhodné zvolit odlišné interní/externí konektory. Místo USB-B použít menší USB-C, místo šroubovacích svorkovnic umístit konektory přímo na DPS. Stejně tak je takřka nutné vyměnit hřebínkové konektory za menší, které nelze standardně zrcadlově prohodit.

V rámci úprav DPS je nutné více odsadit popisky od součástek (na desce rev. 0 je vidět překrývání a nedotisknutí vrstvy) a přidat detailnější popisky ke konektorům.

Zařízení by zároveň mohlo být doplněno o baterii a její nabíjecí modul. To by opět pomohlo konkurenceschopnosti zařízení. V souvislosti s baterkou a její výdrží, je pak důležité vyměnit přepínání pomocí relé za součástky s nižší spotřebou. Stejně tak je možné kvůli spotřebě vyměnit LDO regulátor za spínací – ideálně s integrovanými funkcemi pro nabíjení baterie a podobně.

Dále je vhodné zabudovat alespoň v rámci DPS a kódu mikrokontrolérem ovládaný jas displeje – pomocí PWM.

Pro další verze zařízení je také vhodné zvolit odlišný mikrokontrolér. Původní MCU byl vybírán především s ohledem na cenu, po osazení modelu s větší pamětí se však cena neúměrně zvedla a osazený MCU již nemá do budoucna takový význam. V rámci této práce bylo k osazení dražšího MCU přistoupeno především z časových důvodů. V opačném případě by totiž bylo nutné překreslit a nechat znovu vyrobit DPS.

Při změnách na DPS je možné zvažovat i MCU ve větších pouzdrech. Malé pouzdro LQFP-48 nebylo pro vyrobený analyzátor voleno cíleně, roli při výběru hrála opět cena původně vybraného MCU.

V rámci softwarových změn je nutné zcela změnit grafickou knihovnu MCU a celkovou obsluhu GUI analyzátoru. Použité knihovny sice na prováděné operace dostačují, v rámci rozšiřování zařízení o nové funkce by se však stávaly větší a větší přítěží. Pro větší robustnost tedy příště zvolit ideálně LVGL, TouchGFX nebo Nextion – předpokladem však je větší počáteční časová náročnost.

V aplikaci pro počítač se dále mohla nacházet funkce pro editaci příchozího signálu. V případě realizace je nejprve nutné změnit způsob odesílání signálu z PC

do MCU. Nyní je použit protokol DMX512, který však může zanášet příliš velké zpoždění. V režimu editace by bylo vhodné využít stejného protokolu, který se používá pro přijímání dat, pouze v opačném směru. Samotný režim editace by následně ukládal uživatelem modifikované byty a upravované pakety posílal zpět do MCU.

Během realizace práce bylo zvažováno také ukládání DMX512 dat do textového/.csv souboru – ať už přímo na SD kartu v analyzátoru nebo do PC. Vzhledem k povaze protokolu, by bylo nutné implementovat beztrátovou kompresi signálu. V opačném případě bude totiž textový soubor nabývat na velikosti rychlostí zhruba 1,3 MB/s, což ho bude činit hůře zpracovatelným. Základem kódování by bylo zkrácení za sebou jdoucích opakujících se paketů, v rámci jednoho paketu pak například implementace run-length kódování. Byte na začátku paketu by určoval, zda je run-length kódování aktivní a kódovaly by se pouze pakety, jejichž výsledná velikost by nebyla větší než velikost standardního paketu. U velkých světelných show, kde se často mění všechny kanály signálu by totiž dané kódování ke kompresi nepřišlo – nejvýhodněji by se projevilo při ovládání pouze několika kanálů.

Do aplikace pro PC by následně bylo možné naprogramovat dekodér souborů s časovou osou, kde by dané pakety šly samostatně zobrazit, případně jako celek spustit (přehrát).

4 Návrh a realizace programu mikrokontroléru

Jak již bylo v předchozích kapitolách řečeno, hlavní úlohou mikrokontroléru v této práci je příjem a odesílání signálu DMX512, komunikace s PC a obsluhování uživatelských periférií. V jazyce C, v programovacím prostředí STM32CubeIDE byl pro mikrokontrolér vytvořen poměrně jednoduchý program, který se o potřebné záležitosti stará. Celý kód byl ve verzích postupně ukládán i na server GitHub. V následujících sekcích, bude přiblížen kód programu a důvod volby jednotlivých řešení.

4.1 Nastavení mikrokontroléru a periférií

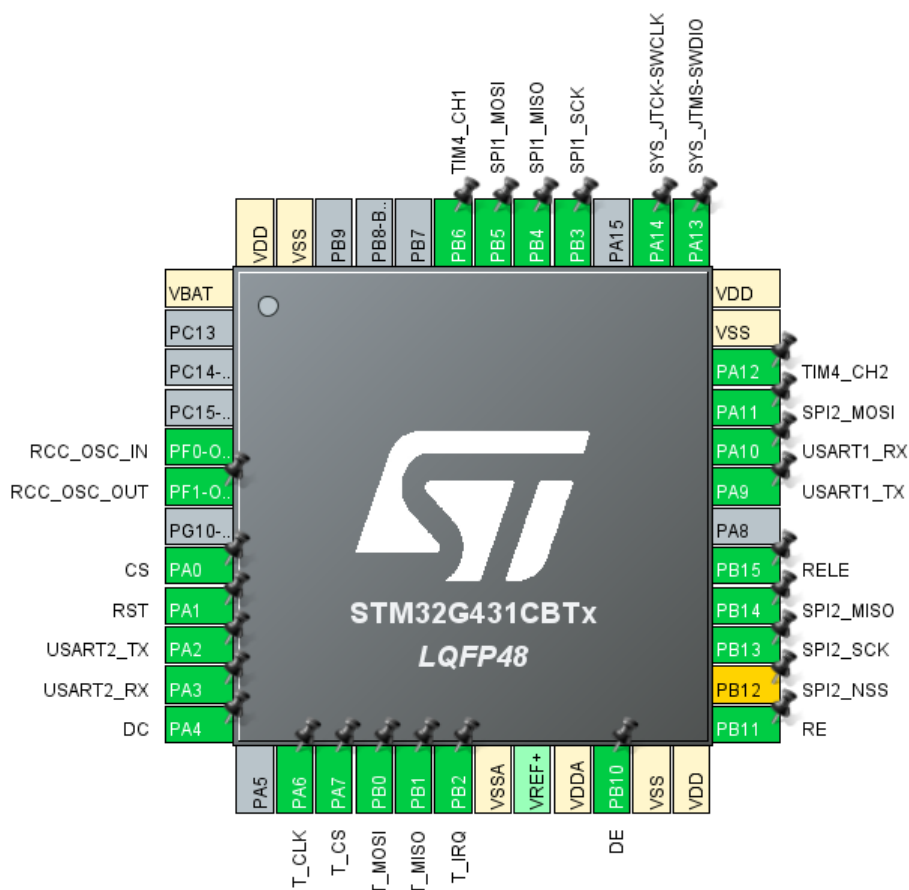
Pomocí aplikace CubeMX bylo provedeno základní nastavení taktu mikrokontroléru. Na desce zařízení je osazen 16 MHz krystal, kterého mikrokontrolér využívá jako zdroje *High Speed Clock* signálu. To umožnilo nastavit takt procesoru na 150 MHz, pro který aplikace automaticky nastavila takt přidružených periférií.

V rámci souboru *main.c* jsou nadefinované funkce pro přepínání režimů analyzátoru, které jsou zobrazené v GUI. Funkce *SwitchToReceiveOnly* a *SwitchToTransmit* pak přepínají použité relé, čímž nastavují zda bude zařízení signál generovat, nebo pouze přijímat.

4.2 Knihovna pro DMX512

Pro příjem a odesílání signálu se zvažovalo použití dostupných open-source knihoven, které pro STM32 existují. Ve snaze více pochopit způsob komunikace a způsob technického řešení analýzy a generování signálu DMX512, bylo přistoupeno k tvorbě vlastní knihovny. Vlastní řešení je založeno na knihovně HAL, a proto by neměl být problém s jeho použitím na odlišných modelech řady STM32.

Přijímání signálu DMX512 je realizováno funkcí *ReceiveToIdle_IT*, při níž mikrokontrolér přijímá data přicházející na UART tak dlouho, dokud na lince nenastane stav idle (logická jedna), čímž mikrokontrolér ukončí přijímání a uloží obsah jednoho celého paketu signálu do pole bytů *receiveBuff*. Po ukončení funkce přijímání se přerušením volá funkce *RxEventCallback*, která opět zahájí příjem signálu. Celé přijímání probíhá pomocí přerušení, a proto probíhá bez blokování hlavní smyčky. Alternativou této funkce je funkce *ReceiveToIdle_DMA*, kde je přijímání realizováno



Obr. 4.1: Nastavení vývodů mikrokontroléru v CubeMX.

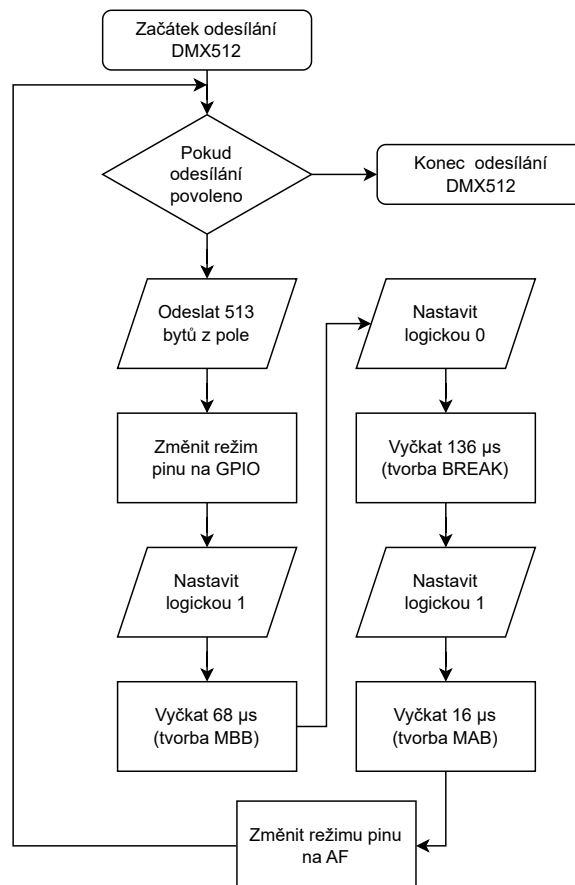
pomocí DMA – data jsou přenášena z UART do paměti na HW úrovni a mikrokontrolér je zpracovává až při jejich celém přijetí. Metoda pomocí přerušení byla pro účely prototypu využita kvůli jednoduchosti.

Odesílání signálu je realizováno velmi podobným způsobem, funkcí *Transmit_IT*. Tato funkce vysílá na vybraný UART data z pole bytů *sendToDMX* a po jejím dokončení je volána funkce *TxCpltCallback*. V rámci této funkce je Tx pin UARTu přepnut z režimu UART (alternate function) na režim GPIO (output), aby mohla být sběrnice několik desítek μs ponechána v klidovém stavu (logická 1) a vytvořilo se tak MBB. Pro načasování MBB je použit hardwarový časovač *TIM7*, nastavený dle vzorce

$$Casovac = \frac{(CounterPeriod + 1)(Prescaler + 1)}{FrekvenceMikrokontroleru} \quad (4.1)$$

na hodnotu zhruba $68 \mu\text{s}$ (CounterPeriod=100, Prescaler=100). Tento časovač $68 \mu\text{s}$ po spuštění vyvolá přerušením callback *PeriodElapsedCallback*, ve kterém je časo-

vač zastaven, na výstup sběrnice nastavena logická nula a spuštěn časovač *TIM6* (přibližně 136 μ s, CounterPeriod=100, Prescaler=200), čímž se vytvoří "SPACE" for BREAK signálu. Časovač *TIM6* je v rámci jeho callbacku zastaven a výstupní pin je nastaven na logickou jedna. Spustí se časovač *TIM3* (přibližně 16 μ s, CP=22, PSC=100), čímž se vytvoří MAB a v rámci callbacku *TIM3* se pin přepíná zpět do režimu UART. Tím je odeslán jeden celistvý paket a pomocí *Transmit_IT* je započato odesílání dalšího paketu.



Obr. 4.2: Vývojový diagram odeslání DMX512 z analyzátoru.

Prvních deset paketů signálu DMX512 se odesílá bez použití časovače *TIM7*. Prodleva pro MBB je zde vytvořena pomocí cyklu while, který po určitý čas blokuje chod programu. Tímto způsobem byla ošetřena chyba, kdy signál z jedné desky analyzátoru nebyl na druhé desce zjistitelný, pokud spolu desky byly propojené a první paket signálu byl vyslán s pomocí časovače *TIM7*.

Odesílání dat do PC je řešeno opět funkcí *Transmit_IT*, ve které se odesílá obsah pole bytů *sendToPC*, ve kterém je kromě 513 bytů signálu DMX512 fixně uložena také hlavička a patička signálu, viz 3.1.3. Přijímání signálu z PC je pak opět řešeno pakety ve formátu DMX512.

Pro komunikaci s PC se využívá *UART1*, převodník úrovní TTL–RS485 je připojen k pinům *UART2*.

4.3 Knihovna dotykového displeje

Výrobce STM32 nabízí ke svým mikrokontrolérům také přesvědčivé grafické rozhraní a knihovny – TouchGFX. Ty spolu se softwarem umožňují WYSIWYG tvorbu GUI a následně jeho rychlou implementaci do mikrokontrolérů [31]. Kvůli jednoduchosti však byla místo TouchGFX vybrána open-source, komunitou vytvořená knihovna, přímo pro zobrazovací řadič ILI9341 a řadič dotykové vrstvy XPT2046, viz [32]. Na prototypu byly zkoušeny i jiné komunitní knihovny, vyskytovaly se však problémy s dotykovou vrstvou. Pro práci bylo vybranou knihovnu nejprve nutné lehce upravit, nyní je však plně funkční.

Knihovna umožňuje základní grafické úpravy (vykreslení pixelu, linky, obdélníku i textu), má předdefinované základní barvy a dokáže obsluhovat dotykovou vrstvu. Knihovna je obsluhována pouze pomocí hlavního kódu – nevyužívá přerušování ani DMA.

V rámci rozšíření práce by bylo vhodné zakomponovat kalibraci dotykové vrstvy. Ta je v analyzátoru zkalibrována manuálně pro konkrétní displej a s jiným displejem by se mohla lehce lišit. Jako řešení se nabízí dvoubodová kalibrace (kliknutí do rohu s nulovými souřadnicemi a kliknutí do rohu s maximálními souřadnicemi), případně přesnější, třibodová, viz [33].

4.4 Grafické uživatelské rozhraní zařízení

Odporová dotyková vrstva použitého displeje vyžaduje vyšší přítlak a disponuje poněkud menší přesností než na jakou jsou uživatelé běžně zvyklí z kapacitních dotykových vrstev chytrých telefonů. Aby nemusel být se zařízením používán přiložený plastový stylus, je GUI navrženo pro ovládání dotykem prstu. To znamená především použití velkých a od sebe dostatečně vzdálených tlačítek.

GUI na zařízení je vytvářeno pomocí funkcí knihovny. V rámci nekonečné smyčky programu je detekován platný dotyk na displeji a následně dle souřadnic dotyku provedeny náležité příkazy. Pro vykreslení menu byly vytvořeny vykreslovací funkce *MainMenu1_Horizontal* a *MainMenu2_Horizontal*, ostatní pod-stránky jsou již definovány v hlavní smyčce programu. V rámci pod-stránek se vždy aktualizuje proměnná *menuNav*, která určuje na které stránce se program nachází, kvůli tlačítkům pro návrat zpět, vpřed, i kvůli práci se signálem.

V každé pod-stránce je nejprve obrazovka nejprve překreslena barvou pozadí, následně dle potřeby pod-stránky opětovně vykresleny navigační šipky a následně vykreslen text. Teprve po vykreslení GUI je spuštěna příslušná funkce analyzátoru.

4.5 Možné rozšíření SW analyzátoru

Pro možné rozšíření práce bylo v rámci software analyzátoru počítáno s rozhraním *SPI2* pro čtečku SD karet i s časovačem *TIM4* v režimu encoder, dedikovaným pro obsluhu rotačního enkodéru. V rámci *.ioc* souboru pro CubeMX a zdrojového kódu jsou tyto periferie aktivovány a nastaveny, v této práci se však nepoužívají.

5 Návrh a realizace aplikace pro PC

Hlavním cílem aplikace pro osobní počítač byla možnost zobrazování přijímaných hodnot signálu DMX512, jejich generování, či úprava. Předpokladem je samozřejmě funkčnost na stávajících počítačích, základní grafické rozhraní a jeho přehlednost. Tato kapitola proto bude věnována volbě programovacího jazyka, grafických knihoven a následně popisu vytvořené aplikace, jejich funkcí a způsobu realizace. Celý kód byl ve verzích postupně ukládán i na server GitHub.

5.1 Výběr programovacího jazyka

Pro programování aplikace byl vybírán moderní, objektově orientovaný jazyk, který je vhodný na tvorbu aplikací pro Windows 10 a novější, umožňuje aplikacím fungovat i bez nainstalovaného frameworku, je multiplatformní a jednoduchý pro vývoj.

Z těchto důvodů byl vybrán jazyk C# s frameworkem .NET 6, tzv. LTS (Long-term Support) verze. Dnes je dostupná také nová LTS verze .NET 8, která však vyšla až v průběhu tvorby této práce.

Jedná se o objektově orientovaný jazyk s virtuálním strojem [30]. Jeho velkou výhodou je jednoduchost a rychlost integrace grafického rozhraní, které lze v rámci vývojového prostředí Visual Studio 2022 (i starších) editovat metodou WYSIWYG. Další výhodou je také kvalitní dokumentace na oficiálních stránkách.

V poslední době zažívá C# rozmach v rámci multiplatformnosti. Původní proprietární „.NET framework“ byl postupně nahrazován open-sourcovým, multiplatformním „.NET Core“ a později přejmenován na „.NET“. Právě .NET zcela nahradil ostatní vývojové větve a nyní je primárně podporovaným frameworkem. Dnes je .NET kompatibilní s operačními systémy Windows, Linux, macOS a díky nástrojům Xamarin také s Android a iOS. Díky tomu je možné vytvářet multiplatformní aplikace [34].

Další výhodou jazyka C# je jeho oficiální integrace v rámci herního engine Unity, díky kterému by bylo možné programovat 3D simulace scén, a podobně.

5.2 Výběr grafického rozhraní

Výběr grafického rozhraní byl poměrně náročný. Jen v rámci frameworku .NET existuje několik oficiálně podporovaných grafických knihoven. Do výběru byly přidány i některé komunitní knihovny, které byly lukrativní díky multiplatformnosti. .NET 6 nabízí grafické knihovny:

- Windows Forms,
- Windows Presentation Foundation (WPF),

- WinUI 2 v rámci Universal Windows Platform (UWP),
- WinUI 3 v rámci Windows App SDK,
- .NET MAUI,

a dále lze používat neoficiální, například:

- Avalonia UI,
- Uno.

Pro práci byl vybrán Windows Presentation Foundation, který syntaxí navazuje na Windows Forms. Narozdíl od většiny zmíněných GUI knihoven není multiplatformní (určen pro OS Windows), ale vzhledem k povaze práce se nejedná o problém. Velká část funkcionalit a syntaxe navíc vychází z Windows Forms, WPF je však doplněno o možnost editovat grafické rozhraní v kódu XAML a o vykreslování pomocí DirectX [35].

5.3 Popis aplikace

Hlavní okno aplikace se skládá z části, která zobrazuje data DMX512 paketů (střed) a z části, která slouží k ovládání komunikace (pravý sloupec ovládacích prvků). Nahoře je pak aplikace vybavena menu, ve kterém lze tlačítkem *Rozložení* změnit způsob zobrazení dat DMX512, případně zobrazit informace o aplikaci.

Tlačítka *Start* a *Stop* slouží k zahájení či ukončení komunikace. To je vždy vázáno k vybranému portu v sekci *Port* a k vybrané funkci v sekci *Funkce*. Pokud je například vybrán port COM5 a nastavena funkce přijímání signálu, bude po zmáčknutí tlačítka *Start* zahájeno přijímání a zobrazování dat DMX512 ze zařízení COM5 (typicky USB–Serial převodníku). Přijímání bude aktivní dokud nebude ručně vypnuto tlačítkem *Stop*.

Základním rozložením hlavního okna pro zobrazování přijímaných dat je *Matice kanálů*. Rozložení je aktivováno se spuštěním aplikace a je zobrazeno na obrázku 5.1. Toto rozložení přidává do hlavního okna sadu 65 textových oken, kde každé z nich představuje hodnotu příslušného kanálu (bytu) signálu DMX512, číslovaného od nuly. První textové okno vlevo nahoře je byte *Start code* a po něm směrem doprava následují okna reprezentující další byty, respektive kanály. První okno za *Start code* tak odpovídá prvnímu kanálu (bytu), druhé druhému, a tak dále. Tlačítka \leftarrow *Zpět* a *Vpřed* \rightarrow je možné přepínat mezi 8 stránkami, každá zobrazuje sadu 64 v pořadí následujících textových oken (65 oken obsahuje pouze první stránka kvůli nultému bytu). To umožňuje ovládat/sledovat všech 513 datových bytů.

Při analyzování zpráv na sběrnici jsou textová okna uzamčena pro zápis (označena našedivělou barvou) a pouze zobrazují přijaté pakety. V pravém dolním rohu aplikace je zároveň zobrazeno, kolik z přijímaných paketů nebylo rozpoznáno a tedy

ani zobrazeno. Ihned po spuštění přijímání může být ztracen jeden nebo dva pakety, viz 5.4.1.

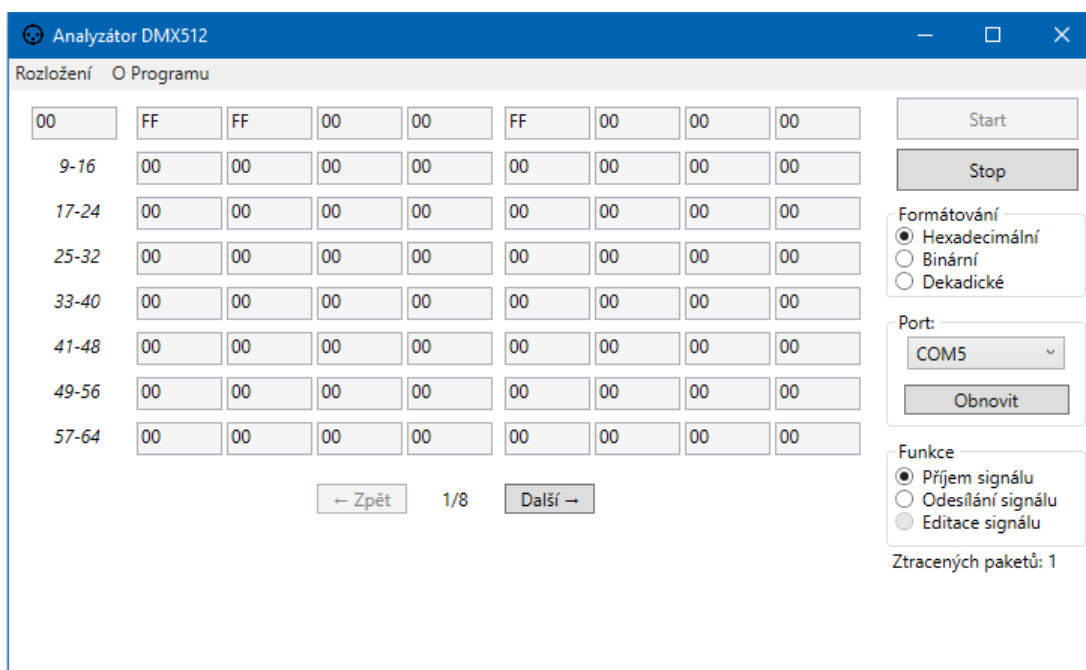
Funkce tohoto rozložení je uživatelsky přívětivá i pro generování zpráv. Po zahájení komunikace je každá změna v libovolném textovém poli okamžitě odesílána na výstup, uživatel tak může sledovat, jak se hodnoty mění už v průběhu zadávání.

Druhým rozložením hlavního okna je zobrazení po jednotlivých kanálech. Zde lze v levém textovém okně samostatně vybrat kanál 0–512 čímž se v pravém okně zobrazí jeho aktuální hodnota. V pravém textovém okně je také možné ručně zadat hodnotu kanálu a stisknutím tlačítka *Potvrdit odesílání* ji odeslat na výstup.

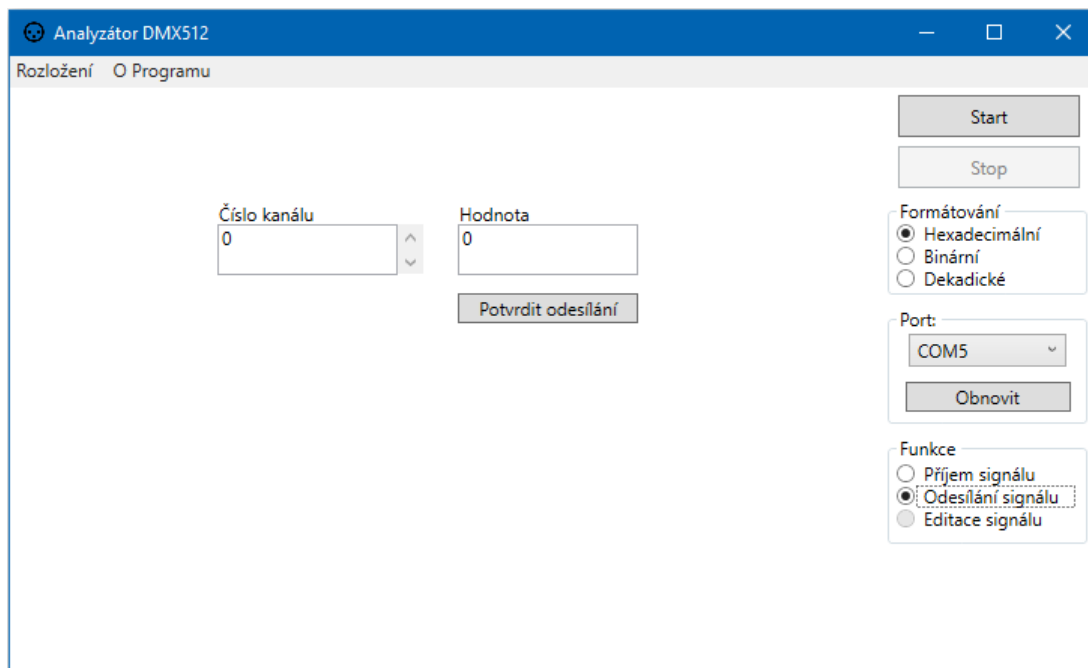
V režimu odesílání zůstávají odesílaná data uložena pro každý port zvlášť. Mezi porty (převodníky) lze proto přepínat a sledovat hodnoty které právě vysílají, případně je měnit. Generovaná data se odesílají kontinuálně (z podstaty protokolu DMX512 by odeslání jednoho paketu nemělo smysl) a vždy korespondují s hodnotami, které jsou viditelně zapsaná v textových boxech.

Aplikace dále umožňuje změnu formátování zobrazovaných dat. V nabídce je hexadecimální, dekadický a binární tvar. V případě připojení převodníku za běhu aplikace, lze obnovit seznam portů tlačítkem *Obnovit* v sekci *Port*.

Najetím myši na některé prvky v hlavním okně, lze zobrazit nápovědu pro daný prvek (objekt).



Obr. 5.1: Aplikace analyzátoru pro osobní počítač, rozložení matice kanálů.



Obr. 5.2: Aplikace analyzátoru pro osobní počítač, rozložení jednotlivých kanálů.

5.4 Vnitřní struktura aplikace

V rámci kódu je aplikace členěna do několika tříd. Nejdůležitější třídou je třída *Protocol*, ve které je definováno přijímání a odesílání signálu DMX512. K tomu využívá jmenného prostoru (namespace) *System.IO.Ports* a jeho třídy *SerialPort*, pro obsluhu sériové komunikace.

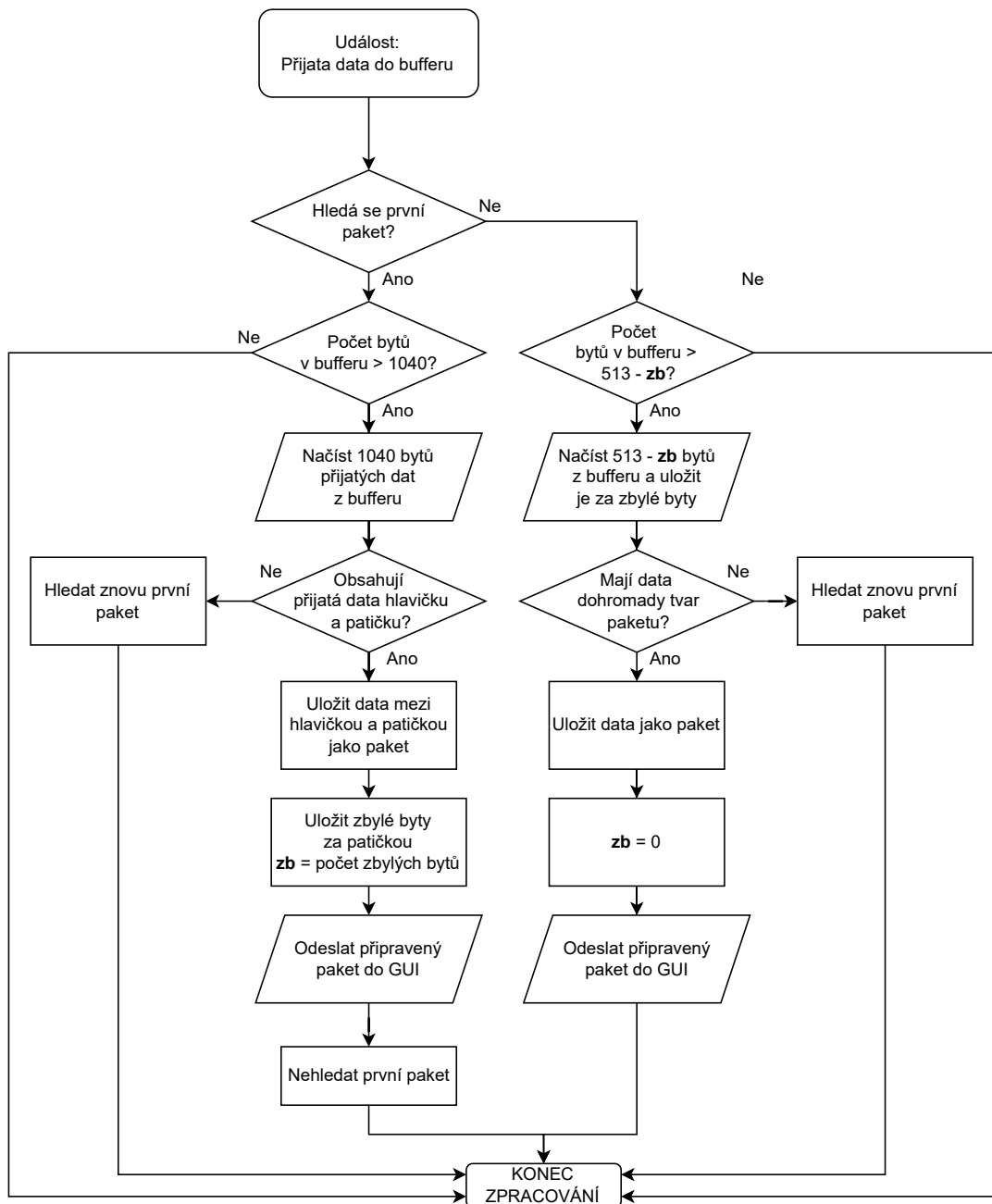
Instance třídy *Protocol* se vytváří samostatně pro každý aktivní COM port, který aplikace nalezne. Díky tomu je možné používat jednotlivé USB–Serial převodníky zcela samostatně (viz 5.3).

5.4.1 Přijímání zpráv

Analýza DMX512 je řešena vlastním protokolem, který byl popsán v kapitole 3.1.3. Program tedy přijímá datový tok na UART, ve kterém průběžně podle předem stanovených bytů – hlavičky a patičky – rozpoznává paket signálu DMX512. Algoritmus přijímání je zobrazen ve vývojovém diagramu 5.3.

Skok do přijímacího algoritmu je v kódu řešen událostí *DataReceived* ze třídy *SerialPort*, ke které je přihlášena asynchronní metoda *SerialPort_DataReceived*. V té se spouští zmíněný algoritmus jako *Task* s parametrem *await*.

Po úspěšném přijmutí paketu je voláno *packetReceivedCallback?.Invoke(packet)*,



Obr. 5.3: Vývojový diagram příjmacího algoritmu v aplikaci.

akce která v instanci *MainWindow* spustí metodu *OnPacketReceived*, jenž daty z předávaného pole *packet* obnoví obsah právě zobrazené stránky.

Podobně funguje i volání *packetDropped?.Invoke(packetDropped)*, které v GUI (*MainWindow*) obnovuje informaci o počtu nerozpoznaných paketů během přijímání. Kvůli prodlevám v rámci vnitřní struktury jazyka a frameworku C#, nemusí být první pakety ihned po začátku komunikace rozpoznány. Pakety se ztratí při za-

vádění komunikace, pravděpodobně vlivem prioritizace úloh v použitých knihovnách .NET. Jedná se o systematickou chybu, kterou je možné vzhledem k povaze protokolu DMX512 zanedbat (protokol odesílá pakety s frekvencí až 44,12 Hz). Bylo by možné ji řešit pozdržením spuštění přijímání od uživatelské interakce s GUI, nebo první pakety jednoduše nazapočítávat do ztracených paketů. Takto je však počítadlo objektivnější, což je u analyzátoru žádoucí – analýza je započata „okamžitě“ po stisku tlačítka *Start* a „daní“ za to jsou právě ztracené pakety.

Algoritmus pro příjem paketů se řídí konstantní proměnnou *packetSize*, deklarovanou a inicializovanou na začátku kódu na 520 (bytů). V případě rozšiřování paketů vlastního protokolu o další byty, je možné změnit pouze tuto hodnotu, algoritmus je pro to přizpůsoben. Je však nutné počítat s tím, že první dva byty a poslední dva byty paketu jsou vždy určeny pro hlavičku a patičku, a že 513 datových bytů je zarovnáno vždy bezprostředně za hlavičku signálu – rozšiřuje se tedy prostor od 514 bytu paketu dále. Z toho také vyplývá, že *packetSize* by neměl být menší než 517.

5.4.2 Odesílání zpráv

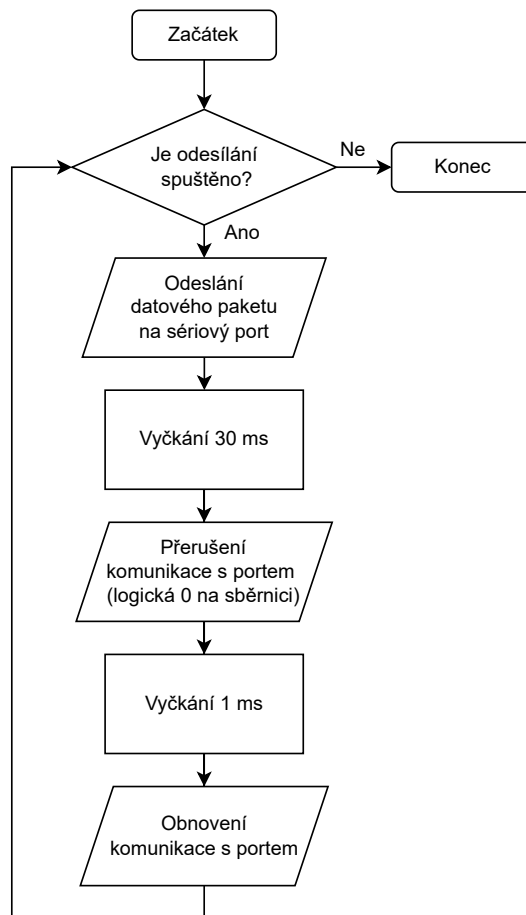
V rámci práce bylo do programu analyzátoru implementováno generování DMX512 zpráv. To funguje pomocí metod *SerialPort.Write* a *SerialPort.BreakState*. Pokud uživatel zahájí odesílání zpráv, je v instanci třídy *Protocol* volána metoda *StartSending*, která nejprve nastaví parametry sériové komunikace (port, baudrate, parita, ...) a následně ve smyčce volá asynchronní metodu *Send* s parametrem *await*. Metoda *Send*, která je návratového typu *Task*, se díky tomu provádí asynchronně a během odesílání zpráv tak nedochází k zastavení grafického uživatelského rozhraní.

Průběh odesílání lze vidět ve vývojovém diagramu 5.4. Data odesílaného paketu jsou v rámci programu uložena v poli bytů *toSend*, počáteční inicializované hodnoty jsou nulové. Uživatelská data do pole zapisuje až GUI metodami *SendHex*, *SendDec* a *SendBin*. Odesílací cyklus mezitím neustále běží, uživatelem změněné byty se tedy propíší na výstup s nejbližším odesláním.

Zatímco při přijímání zpráv se data zobrazují pouze ve zvoleném režimu a vybraném aktivním portu, v režimu odesílání zpráv se odesílají data všech aktivovaných portů.

5.4.3 Obsluha GUI

V aplikaci se dále nachází třídy, určené k ovládání grafického rozhraní (GUI). Chování tříd je obsaženo v souborech s příponou *.cs*, jejich grafické rozhraní pak ve stejnojmenných souborech *.XAML*.



Obr. 5.4: Vývojový diagram odesílacího algoritmu v aplikaci.

Jak již bylo zmíněno, aplikace je navržena tak, aby v ní bylo možné měnit zobrazení přijímaných a odesílaných dat. Proto je obsah GUI rozdělen do třídy *MainWindow*, kde je naprogramováno chování ovládacích prvků hlavního okna, které se se změnou rozložení nemění (Start, Stop, zvolené funkce, ...), a do tzv. stránek (objektů *Page*), které obsahují měnící se objekty zobrazující data. Díky tomuto způsobu je možné nezávisle na ostatním kódu vytvářet další nová rozložení analyzovaných dat.

Načtení dat do rozložení „matice kanálů“ je v přijímacím režimu závislé na callbacku z objektu *Protokol*. Obsah zobrazených textboxů se automaticky obnovuje s každým přijatým paketem (i v případě, že přijatý paket původní vypsanou hodnotu nezměnil). Chování tohoto rozložení je definováno ve třídě *TextBoxPage*. Zde je také specifikováno formátování přijímaného signálu. Režim generování zpráv je založen na událostech *TextChanged* jednotlivých textboxů. Při změně obsahu textboxu uživatelem se jeho nová hodnota odešle do právě vybraného objektu *Protocol* (dle

zvoleného COM portu), kde je parsována jako byte v příslušném formátu a následně uložena do pole pro odesílání.

Chování rozložení „jednotlivých kanálů“ je určeno třídou *ListBoxPage*. Funkce přijímání zůstává stejná jako v případě matice kanálů, obnovuje se zde však pouze jeden, zobrazený textbox. V režimu odesílání se pak nevyužívá události *TextChanged* datového textboxu, ale zápis hodnoty do odesílacího pole je potřeba potvrdit tlačítkem *Potvrdit odesílání*.

Vkládání stránek rozložení do *MainWindow* je provedeno pomocí rámečku (objekt *Frame*), který vloží instanci grafického rozhraní zvolené stránky do grafického rozhraní *MainWindow*.

Pro jednodušší přebírání hodnot objektů grafického rozhraní z *MainWindow* do stránek rozložení, je v kódu vytvořena instance třídy *UserSettings* s názvem *userSettings*, která seskupuje všechna data prvků *MainWindow* do jednoho objektu. Instance *userSettings* se pak předává jednotlivým stránkám v přetížení, při vytváření jejich instancí.

V souboru *MainWindow* je vytvořeno rozhraní *IBasePage*, které musí být implementováno třídami všech grafických rozložení stránek. Díky tomu je totiž pomocí castování možné uložit právě zobrazovanou stránku do proměnné *CurrentPage* typu *IBasePage* a následně používat jednotné metody pro různé typy stránek, ovlivňující vždy pouze vybranou stránku v *CurrentPage*. Toto řešení je podobné zavedení zmíněné třídy *UserSettings*. Třídy v jazyce C# však mohou dědit pouze z jedné třídy, což je již u tříd *TextBoxPage* a *ListBoxPage* využito pro zdědění vlastností třídy *Page* ze *System.Windows.Controls*. Z tohoto důvodu muselo být pro realizaci implementováno rozhraní.

Závěr

Cílem bakalářské práce bylo navrhnout hardware analyzátoru DMX512 a zrealizovat software pro zachytávání zpráv. Tato práce se proto v teoretické části věnovala danému protokolu a v části praktické pak řešila návrh celého zařízení a jeho software. V práci byly zmíněny problémy, které se během realizace vyskytly. Byla popsána jejich možná řešení, promyšlena jejich schůdnost a kvalita, případně jejich vliv na zařízení do budoucna.

V rámci práce byl vyroben funkční analyzátor, který dokáže přijímat a zobrazovat data přenášená na rozhraní DMX512 a umožňuje generovat libovolné zprávy. Zařízení lze ovládat pomocí dotykového displeje a používat ho tak samostatně, nebo připojené k počítači. Analyzátor využívá vlastního konstrukčního řešení s DPS a je uložen v krabici s vyvedenými konektory. Analyzátor je zároveň připraven na rozšíření o další součásti a funkcionality.

V softwarové části byla vytvořena aplikace analyzátoru pro Windows. Ta umožňuje analýzu a generování DMX512 paketů, výběr komunikačního portu, změnu formátování a změnu uživatelského rozhraní. Lze v ní generovaný signál přijímat a prohlížet nebo odesílat.

V rámci práce byly nastíněny další funkcionality a směry, kterými lze zařízení dále rozvíjet, včetně jejich technických řešení.

Literatura

- [1] *Stairville BSW-350 LED User Manual*. Online, PDF. THOMANN GMBH. Thomann. ©2023. Dostupné z: https://images.static-thomann.de/pics/atg/atgdata/document/manual/549467_c_549467_en_online.pdf. [cit. 2023-12-09].
- [2] LIGHTING CONTROLS ASSOCIATION. *Introduction to Lighting Controls*. Online. Lightin Controls Association. ©2023. Dostupné z: <https://lightingcontrolsassociation.org/2017/07/21/introduction-to-lighting-controls/>. [cit. 2023-12-09].
- [3] ENTERTAINMENT SERVICES AND TECHNOLOGY ASSOCIATION [ESTA]. ANSI E1.11-2008 (R2018), *American National Standard ANSI E1.11 — 2008 (R2018) Entertainment Technology-USITT DMX512-A Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories*. R2018. 2018.
- [4] HOWELL, Wayne. *Control Freak A real world guide to DMX512 and Remote Device Management*. Entertainment Technology Press, 2008. ISBN 9781904031550.
- [5] WYMAN, Michael a BEST, Christopher. *Basic Operation of UART with Protocol Support*. Online. MICROCHIP TECHNOLOGY INC. Microchip Docs. 2018. Dostupné z: <https://onlinedocs.microchip.com/pr/GUID-167CA20A-2C0F-4CBC-A693-9FD032B9B193-en-US-1/index.html?GUID-39D5E45F-A0F3-4832-AFEC-DD769377CA0A>. [cit. 2023-11-06].
- [6] MICROCHIP TECHNOLOGY INC. *DMX-512 Protocol*. Online. MICROCHIP TECHNOLOGY INC. Microchip Docs. ©2018. Dostupné z: <https://onlinedocs.microchip.com/pr/GUID-19E32404-10A3-4F82-A4B0-9332290B918C-en-US-1/index.html?GUID-5E934019-999F-4B41-B3E4-4D8D3078C0C1>. [cit. 2023-11-20].
- [7] MA LIGHTING INTERNATIONAL GMBH. *GrandMA3 full-size*. Online. MA LIGHTING INTERNATIONAL GMBH. MA Lighting. ©2023. Dostupné z: <https://www.malighting.com/product/grandma3-full-size-4010500/>. [cit. 2023-11-20].
- [8] TELECOMMUNICATIONS INDUSTRY ASSOCIATION [TIA]. ANSI/TIA/EIA-485-A-1998, *TIA/EIA-485-A*. March 1998.

- [9] FAIRHURST, Gorry. *The DMX Physical Layer*. Online. Electronics Research Group University of Aberdeen. 2018. Dostupné z: <https://erg.abdn.ac.uk/users/gorry/eg3576/DMX-phy.html>. [cit. 2023-11-20].
- [10] SOMMER CABLE GMBH. *DMX Binary 434 DMX512; 4 x 0,34 mm²; PVC Ø 7,00 mm; black*. Online, PDF. Sommer Cable. [2023]. Dostupné z: https://shop.sommercable.com/out/media/me_647_datasheet%20540-0051.pdf. [cit. 2023-12-09].
- [11] *Swisson XMT-500 DMX Tester*. Online. THOMANN GMBH. Thomann. ©1996—2024. Dostupné z: https://www.thomann.de/cz/swisson_xmt_500_dmx_tester.htm. [cit. 2024-05-23].
- [12] ESPRESSIF SYSTEMS. *ESP32-S3-WROOM-1 Datasheet v1.3*. Online, PDF. Espressif. ©2023. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf. [cit. 2023-12-09].
- [13] *Espressif Announces the Launch of ESP32 Cloud on Chip and Funding by Fosun Group*. Online. Espressif. ©2023. Dostupné z: https://www.espressif.com/en/media_overview/news/20160907-esp32briefing#:~:text=Sep%20%2C%202016-,Espressif%20announces%20the%20launch%20of%20ESP32%20Cloud%20on%20Chip%20and,MCU%20at%20Shanghai%20Parkyard%20Hotel.. [cit. 2023-12-09].
- [14] *FreeRTOS Overview*. Online. Espressif. ©2016-2023. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/freertos.html>. [cit. 2023-12-12].
- [15] *ESP32-S3-WROOM-1-N4R2*. Online. Mouser Electronics. ©2023. Dostupné z: <https://cz.mouser.com/ProductDetail/Espressif-Systems/ESP32-S3-WROOM-1-N4R2?qs=sGAEpiMZZMu3sxp5v1qrkr%2F6t0IkXq8jp%252BDC5wIhSw%3D>. [cit. 2023-12-09].
- [16] *STM32G431x6 Datasheet*. Online, PDF. STMicroelectronics. ©2023. Dostupné z: <https://www.st.com/resource/en/datasheet/stm32g431c6.pdf>. [cit. 2023-12-09].
- [17] *STM32G431C6T6*. Online. Mouser Electronics. ©2023. Dostupné z: https://cz.mouser.com/ProductDetail/STMicroelectronics/STM32G431C6T6?qs=uwxL4vQweFMIHkXLHKxSFQ%3D%3D&_gl=1*164ih2n*_ga*dW5kZWZpbmVk*_ga_15W4STQT4T*dW5kZWZpbmVk*_ga_1KQLCYKRX3*dW5kZWZpbmVk. [cit. 2023-12-09].

- [18] *RM0440 Reference Manual*. Online, PDF. STMicroelectronics. ©2023. Dostupné z: https://www.st.com/resource/en/reference_manual/rm0440-stm32g4-series-advanced-armbased-32bit-mcus-stmicroelectronics.pdf. [cit. 2023-12-09].
- [19] *Arduino_Core_STM32*. Online. GITHUB, INC. Github. ©2023. Dostupné z: https://github.com/stm32duino/Arduino_Core_STM32. [cit. 2023-12-12].
- [20] *STM32FreeRTOS*. Online. GITHUB, INC. Github. ©2023. Dostupné z: <https://github.com/stm32duino/STM32FreeRTOS>. [cit. 2023-12-12].
- [21] *What is Arduino?* Online. Arduino. ©2023. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction>. [cit. 2023-12-09].
- [22] N9WXU. *Comparing raw pin toggling speed between Arduino platforms*. Online. In: Microforum. ©2019. Dostupné z: <https://www.microforum.cc/blogs/entry/48-comparing-raw-pin-toggling-speed-between-arduino-platforms/>. [cit. 2023-12-09].
- [23] *Blue Pill*. Online. STM32-base. Dostupné z: <https://stm32-base.org/boards/STM32F103C8T6-Blue-Pill.html>. [cit. 2023-12-09].
- [24] POSCH, Maya. *STM32 Clones: The Good, The Bad And The Ugly*. Online. Hackaday. 2020. Dostupné z: <https://hackaday.com/2020/10/22/stm32-clones-the-good-the-bad-and-the-ugly/>. [cit. 2024-05-20].
- [25] *STM32F103C6T6 STM32F103C8T6 ARM STM32 Minimum System Development Board Module For Arduino*. Online. ©2010-2023. Dostupné z: https://vi.aliexpress.com/item/1005003169244854.html?spm=a2g0o.productlist.main.3.1841tIwhtIwhwk&algo_pvid=ac8f71ec-fbc8-4136-8997-9341ee63ea08&algo_exp_id=ac8f71ec-fbc8-4136-8997-9341ee63ea08-1&pdp_npi=4%40dis%21USD%211.16%211.16%21%21%211.16%21%21%402101efd717021574936352105ea214%2112000024471098728%21sea%21CZ%21818513198%21&curPageLogUId=xxt4n8MPoxK1. [cit. 2023-12-09].
- [26] *OPEN DMX USB*. Online. Enttec. ©2023. Dostupné z: <https://www.enttec.co.uk/en/product/dmx-usb-interfaces/open-dmx-usb/>. [cit. 2023-12-09].
- [27] *FreeStyler DMX*. Online. The DMX Wiki. 2023. Dostupné z: https://www.thedmxwiki.com/dmx_control_software/freestyler_dmx. [cit. 2023-12-09].
- [28] AMERICAN ZETTLER, INC. *AZ822*. Online. American Zettler, Inc. 2/1/16n. 1. Dostupné z: <https://www.azettler.com/pdfs/az822.pdf>. [cit. 2024-05-22].
- [29] LITTLEFUSE, INC. *SM712 Series 600W Asymmetrical TVS Diode Array*. Online. Littelfuse. ©2022, 08/22/19. Dostupné z: <https://www.littelfuse.com/media?>

resourcetype=datasheets&itemid=8313a28c-8802-4d47-a2a7-e30b5b1f67d8&filename=littelfuse-tvs-diode-array-sm712-datasheet. [cit. 2024-05-22].

- [30] *Lekce 1 - Úvod do C# a .NET frameworku.* Online. Itnetwork.cz. ©2023. Dostupné z: <https://www.itnetwork.cz/csharp/zaklady/c-sharp-tutorial-uvod-do-jazyka-a-dot-net-framework>. [cit. 2023-12-09].
- [31] *TouchGFX Documentation.* Online. TouchGFX. ©2023. Dostupné z: <https://support.touchgfx.com/docs/introduction/welcome>. [cit. 2023-12-09].
- [32] martnak. *STM32-ILI9341.* Online. GITHUB, INC. Github. ©2023. Dostupné z: <https://github.com/martnak/STM32-ILI9341>. [cit. 2023-12-12].
- [33] FANG, Wendy a CHANG, Tony. *Calibration in touch-screen systems.* Online, PDF. Texas Instruments. ©2023. Dostupné z: https://www.ti.com/lit/an/slyt277/slyt277.pdf?ts=1702154470265&ref_url=https%253A%252F%252Fwww.google.com%252F. [cit. 2023-12-09].
- [34] *Co je .NET? Úvod a přehled.* Online. Microsoft. ©2023. Dostupné z: <https://learn.microsoft.com/cs-cz/dotnet/core/introduction>. [cit. 2023-12-09].
- [35] *Průvodce pro desktop (WPF .NET).* Online. Microsoft. ©2023. Dostupné z: <https://learn.microsoft.com/cs-cz/dotnet/desktop/wpf/overview/?view=netdesktop-6.0&preserve-view=true>. [cit. 2023-12-09].

Seznam symbolů a zkratek

ANSI	American National Standards Institute
DMA	Direct Memory Access
DPDT	Double Pole, Double Throw
DSP	Digitální signálový procesor
ESTA	Entertainment Services and Technology Association
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer
HW	Hardware
IoT	Internet of Things
LDO	Low-dropout
LSB	Least significant bit
LTS	Long-term Support
MAB	"MARK" After BREAK
MBB	"MARK" Before BREAK
PWM	Pulse Width Modulation
RDM	Remote Device Management
SAI	Serial Audio Interface
SMD	Surface Mount Device
SW	Software
SWD	Serial Wire Debug
THT	Through-hole technology
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
UCPD	USB Type-C Power Delivery

USITT United States Institute for Theatre Technology

WYSIWYG What You See Is What You Get

Seznam příloh

A	Obsah elektronické přílohy	65
B	Vnitřní uspořádání analyzátoru	66
C	Schéma analyzátoru	67
D	DPS analyzátoru	70

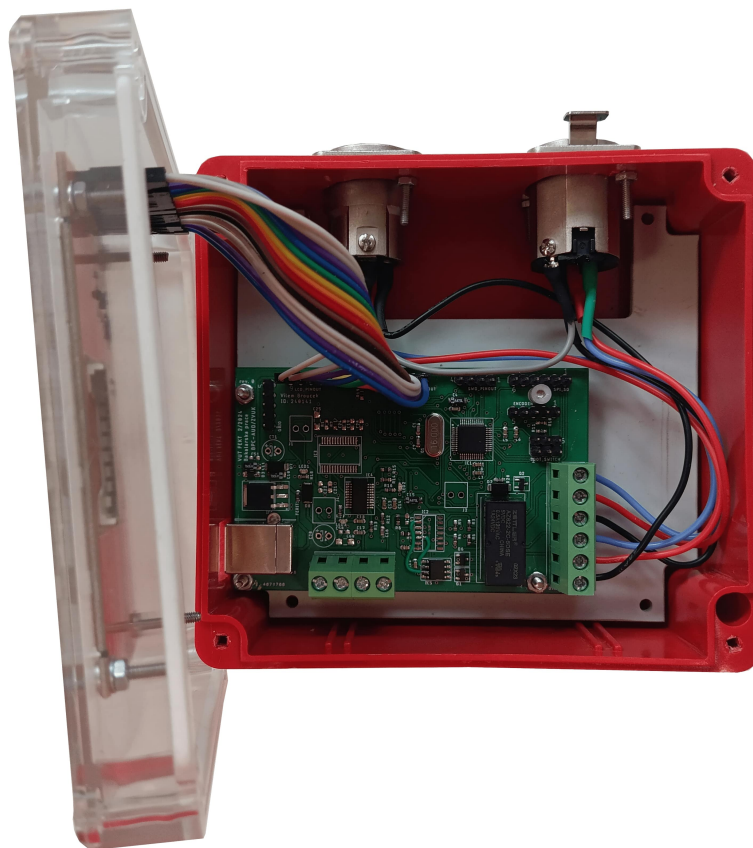
A Obsah elektronické přílohy

Elektronická příloha obsahuje zdrojové kódy programu mikrokontroléru (ve složce *MCU*) a aplikace osobního počítače (ve složce *PC_app*), včetně zkompilovaného *.exe* souboru (ve složce *analyzatorDMX512*).

Aplikace v příloze je kvůli omezené velikosti zkompilovaná pouze pro běh s nainstalovaným .NET 6.0.

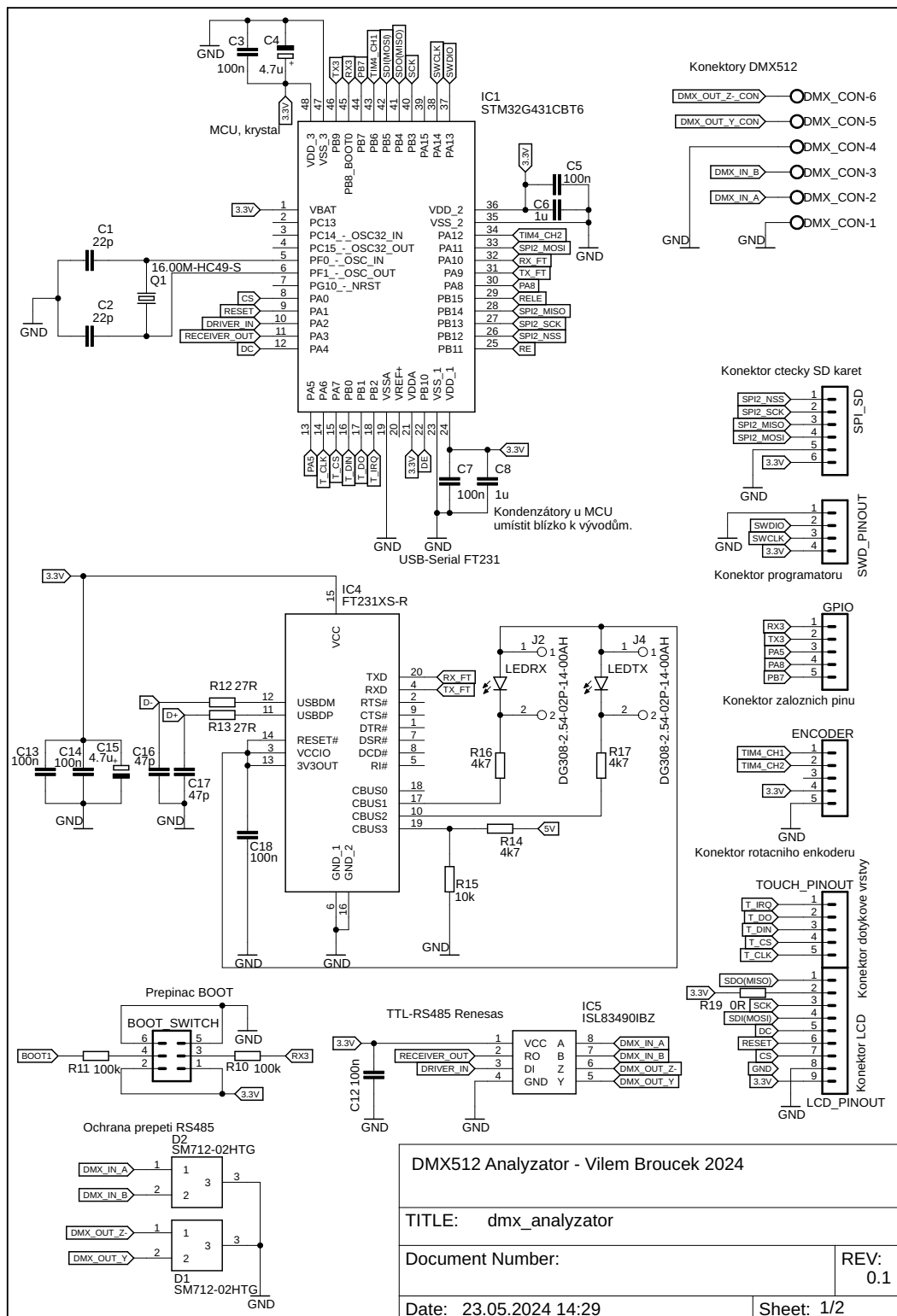
Ve složce *eagle* se nachází soubory schématu a DPS v EAGLE, včetně exportů do PDF – revize 0 (korespondující s vyrobeným zařízením) a revize 0.1 (s opravenými známými chybami). Textový soubor *github* obsahuje odkazy na zdrojové kódy.

B Vnitřní uspořádání analyzátoru

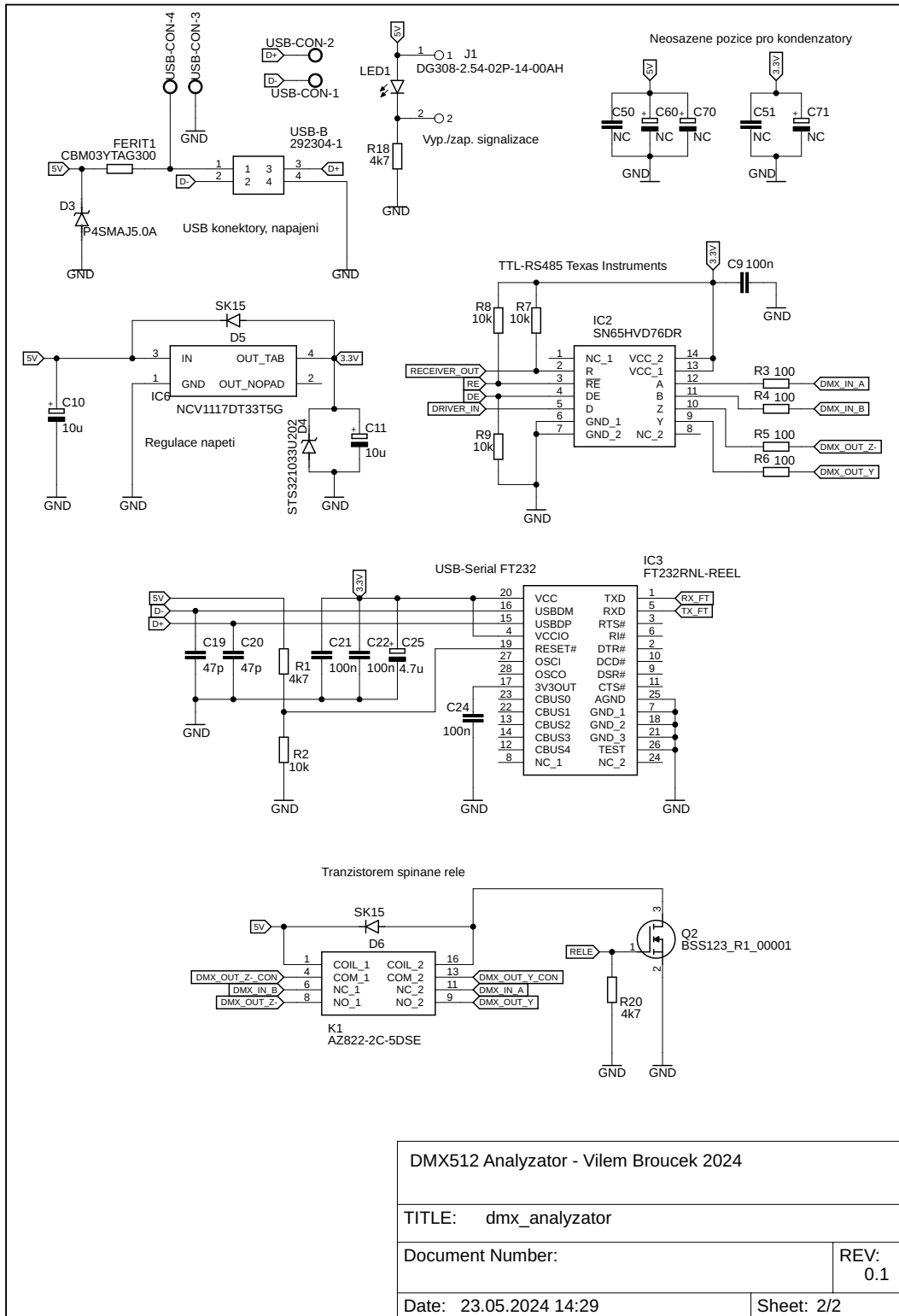


Obr. B.1: Analyzátor bez víka.

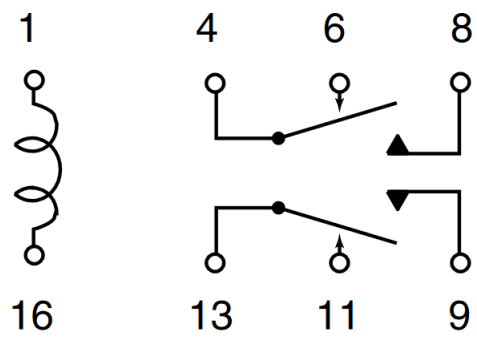
C Schéma analyzátoru



Obr. C.1: Schéma analyzátoru, list 1.

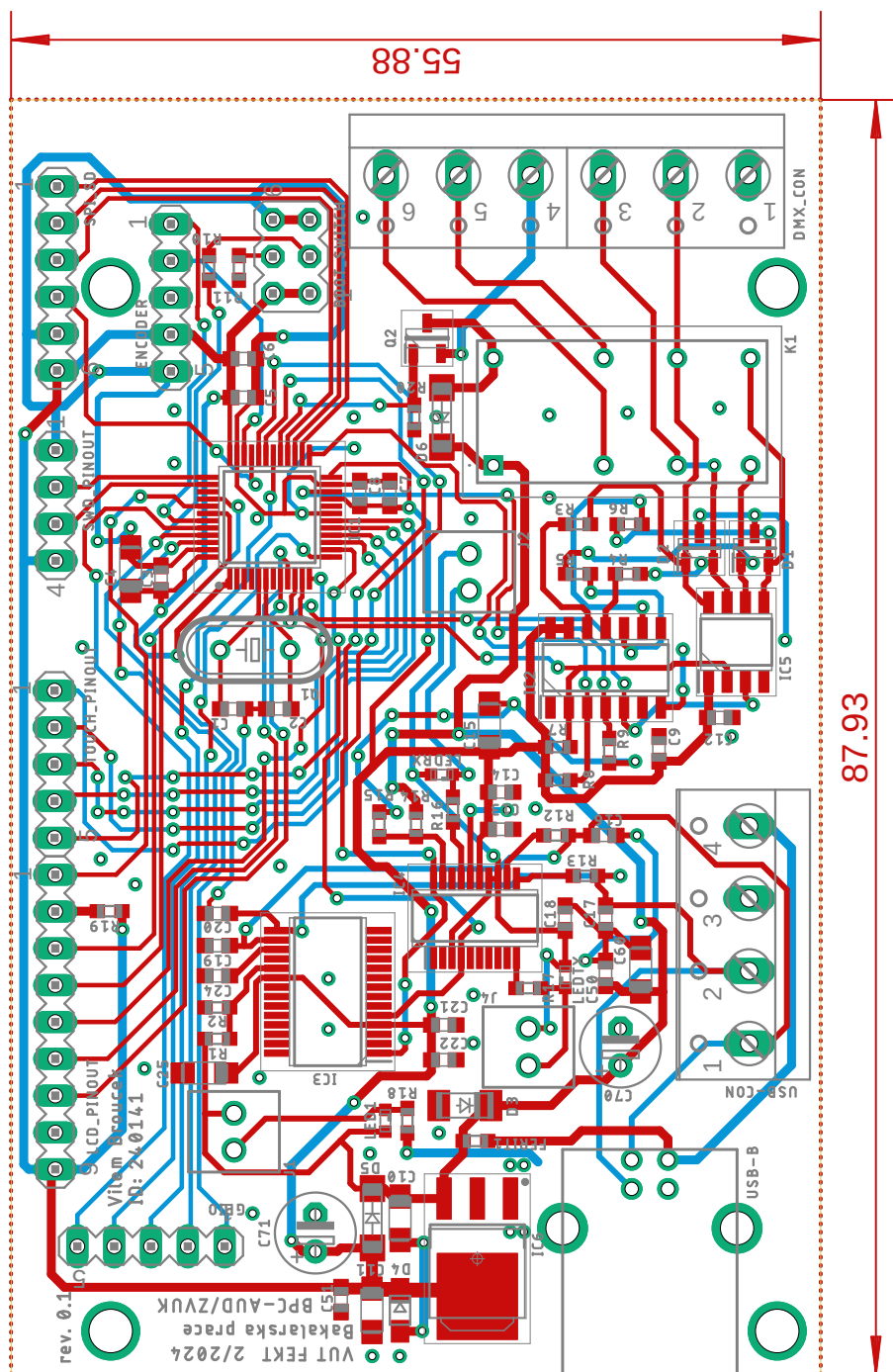


Obr. C.2: Schéma analyzátoru, list 2.

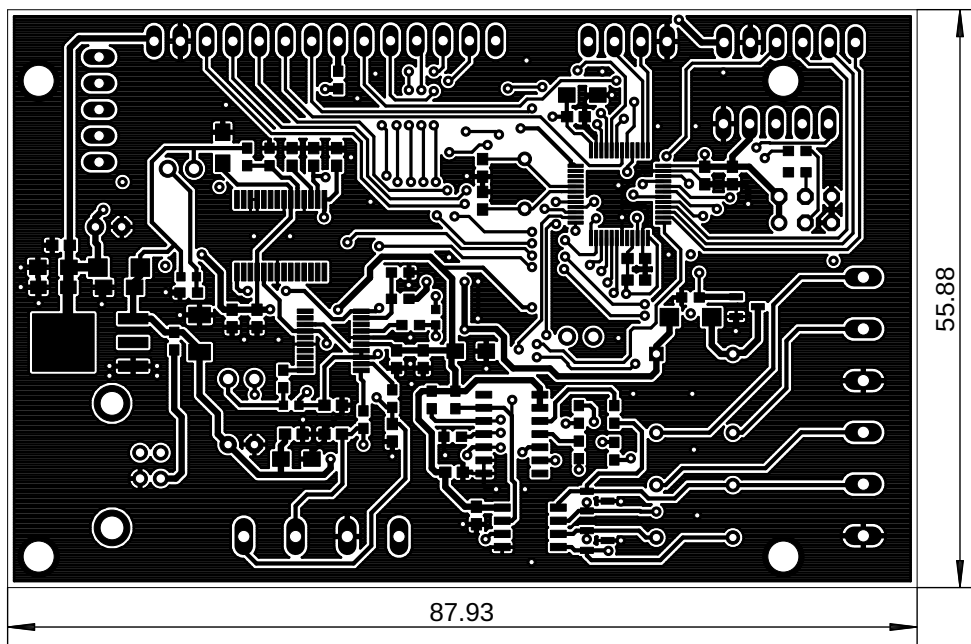


Obr. C.3: Schéma použitého relé [28].

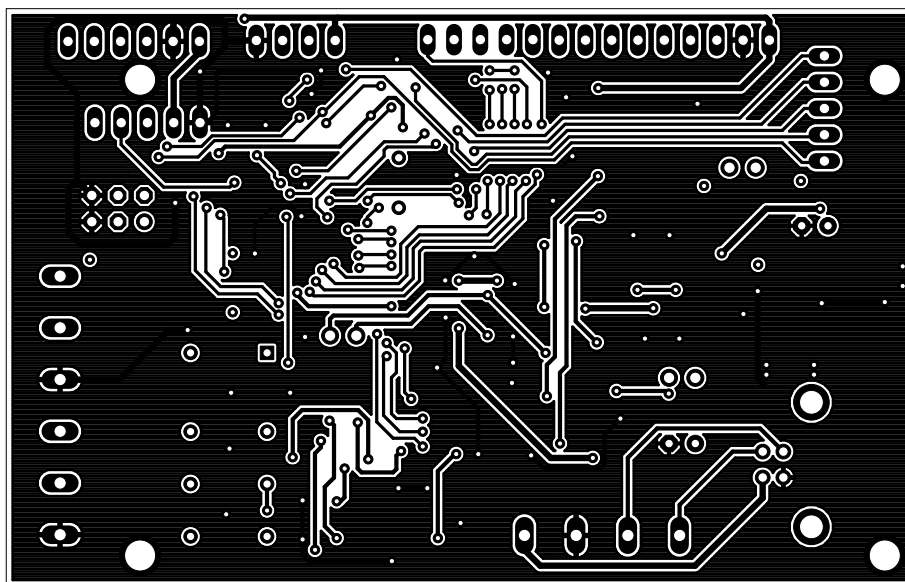
D DPS analyzátoru



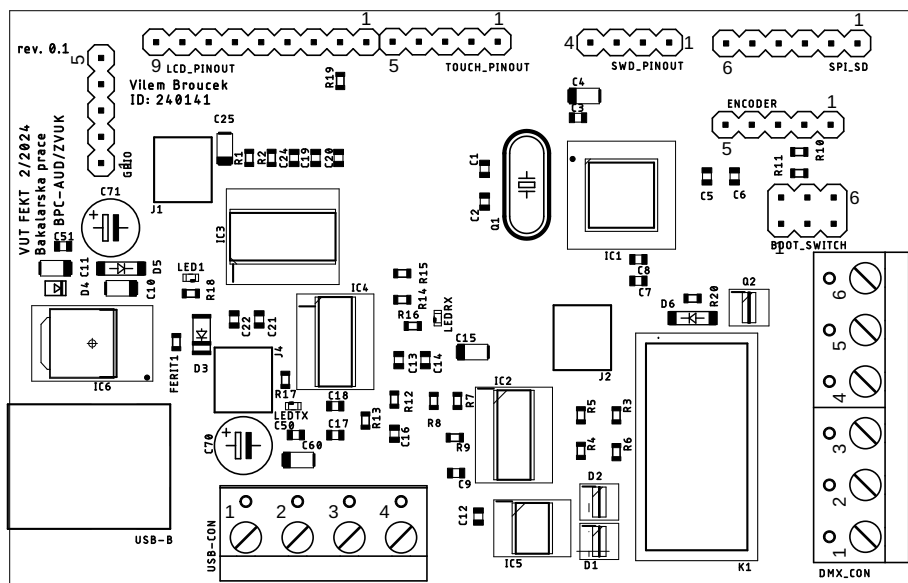
Obr. D.1: DPS – celkový náhled bez zobrazení GND.



Obr. D.2: DPS – vrchní strana, cesty.



Obr. D.3: DPS – spodní strana, cesty.



Obr. D.4: DPS – vrchní strana, potisk.