



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## TESTOVACÍ ZAŘÍZENÍ ELEKTRONIKY PRO VÝSTUPNÍ STANOVIŠTĚ VÝROBY

TESTING STATION FOR PCB DEVICE CHECK

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

**Bc. David Stejskal**

### VEDOUCÍ PRÁCE

SUPERVISOR

**Ing. Jakub Arm, Ph.D.**

**BRNO 2023**

# Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

**Student:** Bc. David Stejskal

**ID:** 211177

**Ročník:** 2

**Akademický rok:** 2022/23

**NÁZEV TÉMATU:**

## Testovací zařízení elektroniky pro výstupní stanoviště výroby

### POKYNY PRO VYPRACOVÁNÍ:

Úkolem je vytvořit systém pro testování vyrobených DPS obsahujících test pointy. K tomu je potřeba vytvořit mechanickou konstrukci, která se na testovanou DPS bude připojovat pomocí pogo pinů, DPS testovacího zařízení na bázi MCU obsahujícího programátor, IO a komunikační rozhraní. Pro zautomatizování testování bude sloužit řídicí desktop aplikace, která bude ovládat a konfigurovat testovací zařízení dle dat z databáze, k čemuž bude využívat standardní USB a Bluetooth rozhraní.

1. Navrhněte koncepci testovacího zařízení.
2. Navrhněte potřebné DPS a realizujte.
3. Navrhněte a implementujte softwarové vybavení.
4. Vytvořte obslužnou desktop aplikaci a zprovozněte celý systém.
5. Vytvořte testovací scénář k danému druhu DPS a otestujte.

### DOPORUČENÁ LITERATURA:

VIRIUS, M. Programování v C++. Grada, 2017. 416 s. ISBN: 978-80-271-0502-1

**Termín zadání:** 6.2.2023

**Termín odevzdání:** 17.5.2023

**Vedoucí práce:** Ing. Jakub Arm, Ph.D.

**doc. Ing. Petr Fiedler, Ph.D.**  
předseda rady studijního programu

### UPOZORNĚNÍ:

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

## ABSTRAKT

Proces testování je nejen ve výrobě neopomenutelnou součástí sloužící k verifikaci vlastností vzhledem ke konkrétním požadavkům na výsledný produkt. Hlavním účelem testování je beze sporu snížení servisních nákladů do budoucna. Určitě není překvapením, že je snahou tento proces co možná nejefektivněji automatizovat, především pro větší série produktů.

Cílem této diplomové práce bylo provést návrh koncepce a následně elektrického, hardwarového i programového řešení testovacího zařízení pro výstupní kontrolu vyrobených desek plošných spojů (DPS). Specifičnost práce spočívá ve vytvoření univerzální testovací DPS, která dokáže programovat a ladit AVR mikroprocesory i čipy od společnosti Espressif, a zároveň je schopna vykonávat automatické testy prostřednictvím tzv. přizpůsobovací DPS. Přizpůsobovací deska je označení pro testovací podstavec, který musí být vždy navržen speciálně na konkrétní produkt (testovaná DPS). Ke komunikaci s periferiemi použitými na této desce využívá testovací DPS výhradně sběrnici I2C, jejíž velkou výhodou je nutnost použití pouhého jednoho páru vodičů. K řízení a vizualizaci testovacího procesu byla vytvořena obslužná počítačová aplikace.

## KLÍČOVÁ SLOVA

Testovací zařízení pro DPS, Kicad 7.0, Microchip Studio, Qt framework, RS-485, SAM D21, obslužná počítačová aplikace, jazyk C, Python, testovací scénář

## ABSTRACT

The testing process is not an indispensable part only of production, it serves to verify the properties with regard to the specific requirements for the final product. The main purpose of testing is, without a doubt, to reduce the service costs in the future. It is certainly no surprise that there is an effort to automate this process as efficiently as possible, especially for larger series of products.

The purpose of this master's thesis is to design a concept and subsequently an electrical, hardware and software solution of a testing device for the output inspection of manufactured printed circuit boards (PCB). The specificity of the work consists in the creation of a universal testing PCB that can program and debug AVR microprocessors and chips from the company Espressif, and at the same time it is able to perform automatic tests through the so-called customizable PCB. The customizable board is a designation for a test stand that must always be designed specifically for a particular product (tested PCB). To communicate with the peripherals used on this board, the test PCB exclusively uses the I2C bus, the great advantage of which is the need to use only one pair of wires. A service desktop application was created to control and visualize the test process.

## KEYWORDS

Testing device for PCBs, Kicad 7.0, Microchip Studio, Qt framework, RS-485, SAM D21, service desktop application, C language, Python, test scenario

STEJSKAL, David. *Testovací zařízení elektroniky pro výstupní stanoviště výroby*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2023, 98 s. Diplomová práce. Vedoucí práce: Ing. Jakub Arm, Ph.D.

## Prohlášení autora o původnosti díla

<b>Jméno a příjmení autora:</b>	Bc. David Stejskal
<b>VUT ID autora:</b>	211177
<b>Typ práce:</b>	Diplomová práce
<b>Akademický rok:</b>	2022/23
<b>Téma závěrečné práce:</b>	Testovací zařízení elektroniky pro výstupní stanoviště výroby

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 diplomové práce panu Ing. Jakobovi Armovi Ph.D. za odborné vedení, přínosné konzultace, trpělivost a podnětné návrhy. Poděkování patří také mé drahé přítelkyni Ing. Petře Vopařilové za její podporu při vypracovávání této závěrečné práce.

# Obsah

Úvod	13
Cíle práce	14
<b>1 Teoretická část práce</b>	<b>15</b>
1.1 Deska plošných spojů	15
1.2 Proces výroby DPS	15
1.2.1 Spektrum vad na DPS	17
1.3 Testování DPS	18
1.3.1 Proces snižování rizik	18
1.3.2 Vizualní kontrola	19
1.3.3 Testování uvnitř obvodu	20
1.3.4 Test funkčnosti DPS	22
1.4 Standardy rozhraní pro sériovou komunikaci	22
1.4.1 TIA/EIA-485 (RS-485)	22
<b>2 Návrh a realizace desek</b>	<b>26</b>
2.1 Přízpůsobovací DPS	27
2.1.1 Funkční bloky elektrické části návrhu	27
2.1.2 Popis funkce a vlastností vybraných komponent	38
2.1.3 Pravidla pro použití TP při návrhu DPS	38
2.1.4 Finální výstup návrhu přízpůsobovací desky	39
2.1.5 Nevyužité části ve finálním návrhu	42
2.2 Testovací DPS	48
2.2.1 Funkční bloky elektrické části návrhu	48
2.2.2 Popis funkce a vlastností vybraných komponent	54
2.2.3 Finální výstup návrhu testovací desky	54
2.3 Rozhraní mezi deskami	56
2.4 Kompletace testovacího zařízení	59
<b>3 Návrh a implementace SW</b>	<b>60</b>
3.1 Microchip Studio	60
3.2 Tvorba ovladačů pro použité periferie	60
3.2.1 Ovladač pro oktálový GPIO expandér	60
3.2.2 Ovladač pro 8-kanálový A/D převodník	62
3.2.3 Ovladač pro EEPROM	63

<b>4</b>	<b>Obslužná aplikace</b>	<b>64</b>
4.1	Softwarové nástroje . . . . .	64
4.1.1	Visual Studio Code . . . . .	64
4.1.2	Qt . . . . .	65
4.1.3	InstallForge . . . . .	66
4.1.4	Poetry . . . . .	67
4.1.5	PostgreSQL . . . . .	69
4.2	Model-View návrhový vzor . . . . .	69
4.3	Okna obslužné aplikace . . . . .	71
<b>5</b>	<b>Testovací scénář</b>	<b>72</b>
5.1	Definice a pojmy . . . . .	72
5.1.1	Testovací případy . . . . .	72
5.1.2	Oblasti testování . . . . .	72
5.1.3	Metriky hodnocení testování . . . . .	73
5.2	Tvorba testovacího scénáře pro testovanou DPS . . . . .	73
5.2.1	Oblasti testování . . . . .	74
5.2.2	Případy testování . . . . .	74
5.2.3	Finální testovací scénář . . . . .	75
	<b>Závěr</b>	<b>78</b>
	<b>Literatura</b>	<b>82</b>
	<b>Seznam symbolů a zkratk</b>	<b>83</b>
	<b>Seznam příloh</b>	<b>86</b>
<b>A</b>	<b>Obsah elektronické přílohy</b>	<b>87</b>
<b>B</b>	<b>Elektrická schémata</b>	<b>88</b>
B.1	Jištění a filtrace napájecího obvodu . . . . .	88
B.2	Programátor modulů ESP32 . . . . .	89
B.3	UPDI – ATmega32U4 . . . . .	90
B.4	Převodník USB na 2x UART – čip FT2232HL . . . . .	91
<b>C</b>	<b>Výrobní výkresy desek</b>	<b>92</b>
C.1	Přízpusobovací DPS – Výrobní výkres s pájecí maskou (vrchní vrstva)	92
C.2	Testovací DPS – Výrobní výkres s pájecí maskou (vrchní vrstva) . .	93



<b>D</b>	<b>Obslužná aplikace</b>	<b>94</b>
D.1	Hlavní okno k ovládnání testování . . . . .	94
D.2	Okno pro nastavení/výběr konfigurace . . . . .	95
D.3	Okno pro obsluhu databáze . . . . .	96
<b>E</b>	<b>Ukázky testovacích skriptů</b>	<b>97</b>
E.1	Hlavní testovací skript . . . . .	97
E.2	Načtení názvu DPS z její konfigurace v EEPROM . . . . .	97
E.3	Změření všech kanálů ADC . . . . .	98
E.4	Aktivace dig. vstupů pro testovanou DPS . . . . .	98

# Seznam obrázků

1.1	Příklad desky plošných spojů . . . . .	16
1.2	Pravidlo 1/10/100/1000 (10×) z ekonomického hlediska . . . . .	19
1.3	Poruchové módy AOI-AXI-ICT procesu . . . . .	20
1.4	POGO piny . . . . .	21
1.5	RS-232 – přenášený datový paket . . . . .	23
1.6	RS-485 – síťové zapojení pro komunikaci v režimu half-duplex . . . . .	25
2.1	Vzájemná konfigurace přizpůsobovací a testovací DPS . . . . .	26
2.2	Blokový diagram přizpůsobovací DPS . . . . .	28
2.3	Řadič z RS-232 na RS-485 . . . . .	29
2.4	Obvod pro ovládání 24 V digitálního výstupu . . . . .	30
2.5	Oddělení referenčních potenciálů logického a výkonového obvodu . . . . .	31
2.6	Oddělení ref. potenciálů logického a výkonového obvodu na DPS . . . . .	32
2.7	Obvod pro ovládání 24 V digitálního výstupu . . . . .	32
2.8	Operační zesilovač s jednotkovým zesílením na vstupu ADC . . . . .	34
2.9	Ochrana vstupů operačního zesilovače proti přepětí . . . . .	34
2.10	Zapojení nevyužitého operačního zesilovače . . . . .	35
2.11	Obousměrný napěťový převodník mezi 3,3 V a 5 V využívající MOSFET . . . . .	35
2.12	3D model přizpůsobovací DPS – pohled shora po úhlem . . . . .	39
2.13	3D model přizpůsobovací DPS – pohled zespodu po úhlem . . . . .	40
2.14	Vrchní vrstva přizpůsobovací DPS – měděná vrstva s potiskem . . . . .	41
2.15	Ochrana sériové komunikace RS-485 . . . . .	41
2.16	Přepětová ochrana digitálních výstupů . . . . .	42
2.17	ESD ochrana – tavná pojistka a TVS . . . . .	43
2.18	Vysokofrekvenční filtr s feritovou perlou . . . . .	44
2.19	Interní zapojení Darlingtonového páru . . . . .	45
2.20	Sériový tranzistorový stabilizátor napětí . . . . .	46
2.21	Simulace sériového tranzistorového stabilizátoru napětí . . . . .	47
2.22	Simulační obvod sériového tranzistorového stabilizátoru napětí . . . . .	47
2.23	DC-DC měnič na +9 V s EMC filtrem na vstupu . . . . .	48
2.24	Blokový diagram testovací DPS . . . . .	49
2.25	Schematické zapojení 4-kanálového USB hubu . . . . .	50
2.26	Schematické zapojení tranzistorů pro automatický programovací režim . . . . .	52
2.27	Schematické zapojení jumperu pro selekci napájecího napětí UPDI a ISP konektoru . . . . .	53
2.28	Doporučené rozložení programovacího konektoru pro ARM SWD/JTAG . . . . .	53
2.29	3D model testovací DPS – pohled shora po úhlem . . . . .	54
2.30	3D model testovací DPS – pohled zespodu po úhlem . . . . .	56

2.31	Vrchní vrstva testovací DPS – měděná vrstva s potiskem . . . . .	57
2.32	Ukázka rozmístění blokovacích kondenzátorů USB hubu . . . . .	57
2.33	Jištění a filtrace napájecího obvodu na DPS . . . . .	58
2.34	Testovací zařízení – propojení testovací a přizpůsobovací DPS . . . .	59
2.35	Testovací zařízení – propojení přizpůsobovací a testované DPS . . . .	59
4.1	PostgreSQL rozšíření pro VSCode . . . . .	65
4.2	Model-View návrhový vzor – přehledový diagram . . . . .	69
4.3	Obslužná aplikace – Měření . . . . .	71
4.4	Obslužná aplikace – Terminál pro RS-485 . . . . .	71
5.1	Vývojový diagram testovacího scénáře . . . . .	76

# Seznam tabulek

1.1	Spektrum vad na DPS . . . . .	18
1.2	Vrstvový model Modbus RTU využívající TIA/EIA-485(232) na fyzické vrstvě . . . . .	23
1.3	Srovnávací tabulka elektrických vlastností RS-232, RS-422, RS-485 . .	25
2.1	Přehled registrů EEPROM . . . . .	36
2.2	Podstatné vybrané komponenty na přizpůsobovací DPS . . . . .	38
2.3	Podstatné vybrané komponenty na testovací DPS . . . . .	55
2.4	D-Sub DC-37 konektor – rozhraní mezi testovací a přizpůsobovací DPS	58

# Úvod

Proces testování je nejen ve výrobní sféře základním předpokladem pro to, aby byla zajištěna co možná nejnižší pravděpodobnost brzké nefunkčnosti a poruchovosti. Je tak možné se setkat s různými formami stanovišť výstupní kontroly skrze všechny oblasti, ve kterých dochází k různým formám generování výsledného produktu.

Při procesu testování je snahou provádět potřebné specifické kroky, díky kterým lze zaručit určitou kvalitu a spolehlivost daného produktu, pokud konkrétním testem úspěšně projde (kritérium *pass/fail*). Aby bylo možné klasifikovat či zhodnotit testovanou vlastnost, musí pro ni být vybráno vhodné testovací kritérium.

K určení stavu ověřované vlastnosti či parametru je vždy zapotřebí konkrétního typu měřicího elementu (senzoru) a následného vyhodnocovacího prvku. Ať už je tímto vyhodnocovacím prvkem člověk nebo sofistikovaný systém, jedná se o neopomenutelnou část stanoviště výstupní kontroly.

Pro otestování vyrobených desek plošných spojů (DPS) bývá v praxi využíváno hned několika technik. Volba se často odvíjí od toho, jak moc sofistikovaným vybavením daný výrobce disponuje a to většinou závisí na množství produkovaných desek. Nejzákladnější technikou bývá měření specifických elektrických napětí v konkrétních místech DPS. Tento přístup je nazýván *In-Circuit Testing* (ICT) a jeho velkou výhodou je poměrně snadná implementace, spolehlivá detekce zkratů/otevřených obvodů nebo možnost ověření důležitých parametrů (elektrický odpor, indukčnost, kapacita) osazených součástek před připojením napájecího napětí.

Tato práce se zabývá kompletním návrhem a následnou realizací testovacího zařízení vyrobených DPS obsahující tzv. testovací piny (*test points*). Testovací zařízení se bude skládat ze dvou desek – testovací a přizpůsobovací. Zatímco testovací DPS bude univerzální a v ideálním případě neměnná, přizpůsobovací DPS (resp. testovací podstavec) bude navržena vždy speciálně na konkrétní verzi testovaného produktu. Tímto přístupem bude docíleno toho, že v případě změny testované DPS (nová verze, změna rozložení testovacích pinů), nebude nutné měnit celý testovací systém.

Jednotlivé konfigurace přizpůsobovacích DPS budou ukládány/načítány do/z vestavěné (*embedded*) paměti EEPROM. Za účelem pohodlného ovládání, konfigurování a vizualizování zautomatizovaného procesu testování bude navržena obslužná počítačová (*desktop*) aplikace.

# Cíle práce

Hlavním cílem této diplomové práce je vytvořit celkový návrh koncepce a elektrického, hardwarové i programového řešení **Testovacího zařízení elektroniky pro výstupní stanoviště výroby**. Nejprve bylo nutné vydefinovat všechny klíčové vlastnosti testovacího systému a současně se zamyslet nad celkovou konfigurací. Vzhledem ke skutečnosti, že systém byl navrhován na produkt, který je ještě stále ve fázi vývoje, bylo nutné absolvovat několik poměrně obsáhlých konzultací s vedoucím této práce. Jeho podnětné návrhy byly velmi přínosné a měly významný vliv na samotný výsledek.

## Byly vydefinovány následující požadavky na testovací zařízení:

- univerzálnost testovací DPS,
- možnost programování a ladění více typů *Microcontroller Unit* (MCU) na testované DPS prostřednictvím vestavěných programátorů ESP-PROG a *Unified Program and Debug Interface* (UPDI) po rozhraní *Universal Serial Bus* (USB),
- možnost programování a ladění MCU (SAM D21) na testovací DPS po rozhraní *Joint Test Action Group* (JTAG)/*Serial Wire Debug* (SWD),
- možnost uložení konfigurací přizpůsobovací DPS na *Electrically Erasable Programmable Read-Only Memory* (EEPROM) komunikující po *Inter-Integrated Circuit* (I2C),
- měřitelnost konkrétních napěťových úrovní (napájení a digitální výstupy) testované DPS s využitím POGO pinů a 8-kanálového *Analog to Digital Converter* (ADC) na přizpůsobovací DPS komunikující po sběrnici I2C,
- schopnost ovládání digitálních výstupů (fungujících až do 40 VDC) na přizpůsobovací DPS s využitím oktálového *General-Purpose Input/Output* (GPIO) expandéru komunikujícím po sběrnici I2C, které budou následně připojeny s využitím POGO pinů a přizpůsobovacích obvodů na digitální vstupy MCU na testované DPS,
- možnost volby konfigurace přizpůsobovací DPS (odpovídající různým testovaným DPS) před spuštěním automatického testu pomocí desktopové aplikace,
- proveditelnost automatického testu podle definovaného testovacího scénáře,
- schopnost komunikace desktopové aplikace s testovací DPS po sériové lince a s testovanou DPS po RS-485 (příkazy a odezvy na ně),
- ovladatelnost celkového testovacího systému pomocí obslužné aplikace,
- funkce generování výsledné testovací zprávy,
- případně možnost zápisu a čtení dat do/z *Structured Query Language* (SQL) databáze.

# 1 Teoretická část práce

Následující podkapitoly a sekce jsou zaměřeny na výběr oblastí, kterým je vhodné porozumět pro pochopení souvislostí, jimiž se zabývá tato závěrečná práce. Z důvodu komplexnosti jsou součástí teoretického úvodu pouze některé oblasti a předpokládá se, že čtenář již zná určité základy této problematiky.

První podkapitola definuje základní vlastnosti, vizuální a funkční stránku DPS. Popisuje typy a technologie součástek používaných při procesu osazování DPS.

V druhé podkapitole je snahou uvést čtenáře do výrobního procesu DPS. Nejprve jsou uvedeny jednotlivé kroky procesu a následně podrobněji popsány. Po přečtení by měla být zřejmá komplexnost tohoto procesu, při kterém může vzniknout různorodé spektrum vad.

Třetí podkapitola pojednává o významu procesu testování DPS, který je úzce spjat s procesem snižování rizik. Dále jsou zde zmíněny nejčastěji používané způsoby testování a jejich principy.

Čtvrtá podkapitola se zabývá sériovými komunikačními standardy, konkrétně především TIA/EIA-485. Následuje srovnání s dalšími variantami tohoto standardu.

## 1.1 Deska plošných spojů

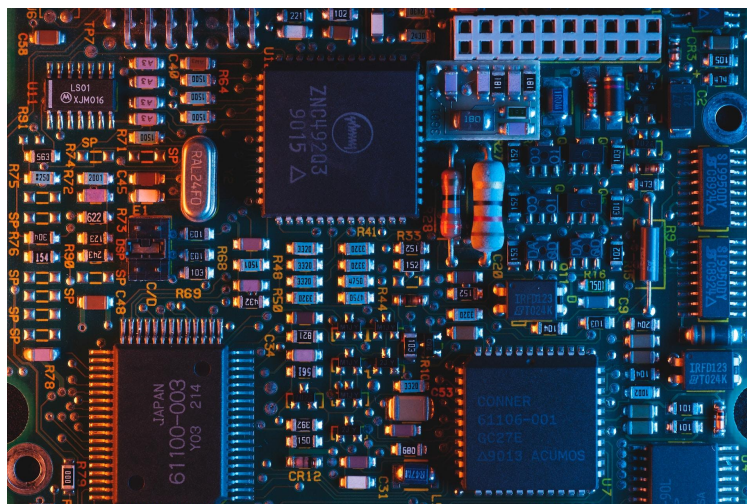
DPS je v elektrotechnice nejrozšířenější médium používané za účelem zprostředkovat pomocí plochých měděných cest elektricky vodivá spojení mezi elektronickými komponenty, které jsou na ni „připevněny“. Specifickým propojením těchto komponent vzniká uzavřený elektrický obvod zajišťující požadovanou funkci, viz Obr. 1.1. Nejčastějšími typy osazovaných komponent jsou *Through-hole Technology* (THT) a *Surface Mount Device* (SMD).

Struktura těchto desek připomíná laminovanou sendvičovou strukturu sestávající z vodivých (měděných) a izolačních (sklolaminátových) vrstev. Typická tloušťka těchto desek je 1,6 mm, přičemž vnější měděné vrstvy (vrchní/spodní) jsou tenké 35  $\mu\text{m}$  (nebo dvojnásobná – 70  $\mu\text{m}$ ) a v případě vícevrstevých (4/6/8-vrstvé) DPS mají vnitřní měděné vrstvy tloušťku běžně 17,5  $\mu\text{m}$  (nebo až 70  $\mu\text{m}$ ).

Vodivého a zároveň mechanického spojení elektronických komponent s vnějšími vrstvami DPS je dosahováno pájením. S vnitřními vrstvami (a navzájem mezi nimi) se vodivé spojení provádí pomocí tzv. prokovů (*vias*) – pokovené průchozí otvory.

## 1.2 Proces výroby DPS

Po úvodu do významu používání a struktury DPS jsou popsány jednotlivé kroky jejího výrobního procesu, který je nazýván *Printed Circuit Board Assembly* (PCBA).



Obr. 1.1: Příklad desky plošných spojů  
(Dostupné z: <https://unsplash.com/photos/jXd2FSvcRr8>)

Komponenty jsou na DPS sestaveny tak, aby zajistily to nejpodstatnější, a tedy její požadovanou funkci. Správné funkčnosti může zamezit i drobná maličkost, jako je například nepřesné osazení malého SMD rezistoru nebo nepřesné useknutí části desky obsahující malou vodivou cestičku. Je tedy velmi důležité, aby byly jednotlivé kroky výrobního procesu provedeny správným způsobem a s dostatečnou přesností. Jednotlivé kroky PCBA procesu jsou:

- nanesení pájecí pasty pomocí šablony,
- rozmístění komponent (*Pick and Place*),
- pájení přetavením (*reflow soldering*),
- inspekce a kontrola kvality,
- vložení a zapájení THT komponent,
- finální kontrola a test funkčnosti.

Pájecí pasta je nanášena na místa, kde se budou následně nacházet vývody jednotlivých elektronických komponent. Tzv. „nanašeč“ rovnoměrně roztírá správné množství pájecí pasty skrze otvory uvnitř nerezové šablony, která je mechanicky pevně spojena s osazovanou deskou. Pájecí pasta je složena z 96,5 % cínu, 3,0 % stříbra a 0,5 % mědi<sup>1</sup>.

Po nanesení pájecí pasty následuje rozmístění SMD součástek na osazovanou DPS. K tomu je využíván tzv. *Pick and Place* robot, jemuž je nutné poskytnout potřebná data (souřadnice X, Y a natočení každé komponenty použité v DPS). Součástky jsou z výchozí pozice (cívka/kotouč (*reel*)) přenášeny pomocí pneumatického principu (podtlaku) a následně umístěny na aplikovanou pastu. Jak postupem času

<sup>1</sup>Převzato a upraveno z: <https://www.pcbcart.com/Uploads/Editor/2017-03-22/...>



technologie vypsěla, tito automatizovaní Pick and Place roboti velmi usnadnili elektrotechnikům práci a dosáhli nesrovnatelně vyšší rychlosti a přesnosti. Navíc jsou schopni pracovat 24/7 bez únavy.

Třetím krokem je pájení přetavením. DPS s nanesenou pastou a osazenými SMD součástkami jsou umístěny na pásový dopravník, kterým jsou následně dopraveny do velké pece rozehřáté na požadovanou teplotu (cca 250 °C), čímž dojde k přetavení pájky. Dalším stanovištěm jsou chladiče, které dříve roztavenou pájku kontrolovaně ochladí a zpevní. Tím vzniká pevný spoj a SMD součástky jedné strany DPS jsou tak od tohoto okamžiku připevněny. Pokud DPS obsahuje SMD součástky z obou stran, je dopsud popsány kroky zpracována nejprve ta strana, která obsahuje menší komponenty.

Po připájení všech SMD komponent je nutné provést kontrolu (*inspection*). Během ní je nutné objevit co nejvíce vad vzniklých při předcházejících krocích. Tato problematika je podrobněji popsána v sekci 1.3.2.

THT komponenty jsou stále často využívány při návrhu DPS, avšak v automatizované výrobě už je snahou od nich upouštět. Jejich nejběžnějšími způsoby pájení jsou manuální nebo automatizované s využitím pájecí vlny (*wave soldering*).

Když už jsou všechny komponenty zapájeny, následuje finální kontrola a test funkčnosti desky. To znamená, že v této fázi už je připojeno napájení a na příslušných testovacích pinech jsou ověřeny hodnoty napětí, proudů, případně správnost digitálního signálu. V případě naměření nepřijatelných hodnot je deska vyřazena a následně opravena nebo zlikvidována podle standardních postupů společnosti. Podrobněji je tato fáze popsána v podkapitole 1.3.

Ve finální fázi PCBA procesu je osazená a otestovaná deska očištěna od zbytků tavidla (*flux*), špíny z prstů i olejových skvrn. K čištění všech těchto typů nečistot je vhodná deionizovaná voda, kterou je DPS vysokým tlakem ostříkována. Následuje vysušení stlačeným vzduchem a potom už lze desku zabalit a odeslat zákazníkovi.

K čištění všech typů nečistot postačí vysokotlaký mycí nástroj z nerezové oceli využívající deionizovanou vodu. Deionizovaná voda nepoškodí obvod PCB. Po umytí se DPS vysuší stlačeným vzduchem.

V předchozích odstavcích, týkajících se PCBA procesu, bylo čerpáno z [16, 13].

### 1.2.1 Spektrum vad na DPS

V Tab. 1.1 je v prvním sloupci uvedeno rozložení vad vyrobených DPS, druhý sloupec vyjadřuje úspěšnost nalezení vad za použití konkrétní detekční metody a ve třetím sloupci je vyjádřeno spektrum vad. Data vycházejí ze statistiky provedené na jedné miliardě zapájených spojů, kterou zhotovil Stig Oresjo, v té době konzultant testovací strategie pro Agilent Technologies, v roce 1999 [12].

Je zřetelné, že převážnou část rozdělení vad tvoří především otevřené obvody, zkratky a kvalita pájení. Společně s nepřesným umístěním komponent jsou jmenované vady poměrně dobře detekovatelné pomocí kamer, a právě proto lze očekávat vysoké procento úspěšnosti nalezení těchto vad optickou inspekcí (viz podsekce 1.3.2). Nicméně to neznamená, že by ostatní metody nemělo smysl používat. Naopak jsou schopné detekovat i ty vady, které optická inspekce nedokáže.

Tab. 1.1: Spektrum vad na DPS [12]

Rozdělení vad	Detekční metody	Spektrum vad
41 % otevřené obvody	92 % optická inspekce	90 % strukturální
20 % zkratky	76 % testování uvnitř obvodu	10 % elektrické
20 % kvalita pájení	69 % test funkčnosti	
8 % umístění		
8 % elektrická		
3 % ostatní		

## 1.3 Testování DPS

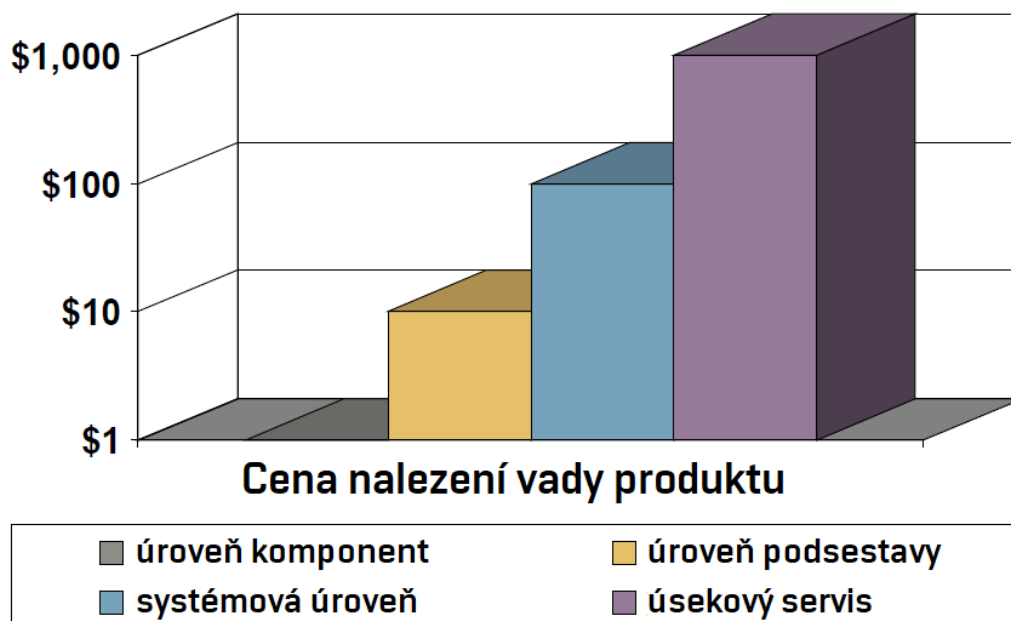
Testování je nedílnou součástí výroby DPS. Zákazník očekává spolehlivé a funkční zařízení, a proto je nutné začít s testováním hned na začátku – při výrobě DPS.

Jako každý proces i testování zabere určitý čas, a ten je snahou co nejvíce redukovat za účelem zkrácení doby výrobního procesu. To již napovídá, že proces testování bývá zpravidla u větších sérií automatizován.

Kdyby se testování neprovádělo, velmi často by nastávala situace, kdy zařízení nefunguje již při doručení zákazníkovi. Tato vada by byla následně objevena při uvádění do provozu (v lepším případě by zákazník otestoval zařízení po jeho obdržení), a takové zjištění lze považovat za velmi nežádoucí a zároveň jedno z nejnákladnějších.

### 1.3.1 Proces snižování rizik

Hlavním smyslem testování je snížit riziko poruchy (*Risk Reduction Process (RRP)*) zařízení pokud možno na minimum. Na Obr. 1.2 je znázorněn graf závislosti ceny za vadu produktu vzhledem k době jejího objevení. Z grafu jasně vyplývá, že čím dříve se chyba podchytí, tím menší jsou náklady za opravu/reklamaci produktu. Například pokud produkt s vadou projde fází testování sestavy až do testování na systémové úrovni, rozdíl v nákladech za stejnou chybu činí \$90, a to zdaleka není zanedbatelná částka při menších či větších výrobních sériích.



Obr. 1.2: Pravidlo 1/10/100/1000 (10×) z ekonomického hlediska  
(Převzato a upraveno z [21])

### 1.3.2 Vizuální kontrola

Budeme-li vycházet z výrobního procesu DPS popsaného v podkapitole 1.2, vizuální kontrole předchází nanášení pájecí pasty, rozmístění SMD komponent a jejich zapájení. Vizuální kontrolu lze v podstatě rozdělit na manuální a optickou/rentgenovou.

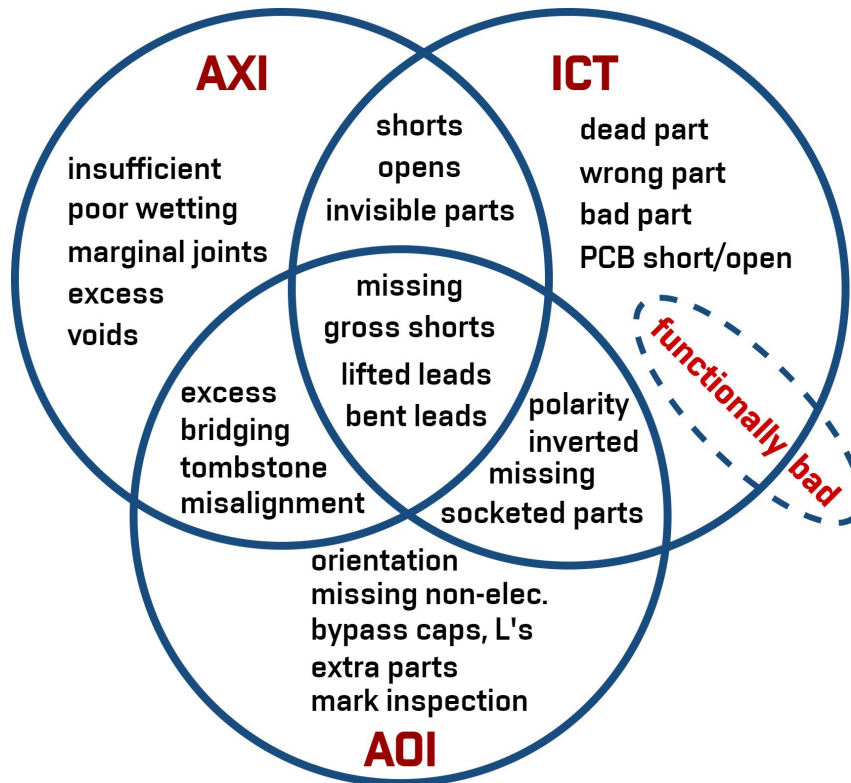
#### Manuální kontrola

Stejně tak jako u procesu rozmísťování komponent (viz podkapitola 1.2) bez pomoci automatického robota, i manuální kontrola (obzvláště malých SMD součástek) prováděná elektrotechnikem je velmi zdoluhavá a při požadavku na dlouhodobě velkou přesnost také velmi náročná. Proto už se od tohoto způsobu dávno upustilo a využívají se výhradně automatizovaní roboti. Manuální kontrola je tedy využitelná spíše pro DPS obsahující THT součástky nebo při malé hustotě komponent.

#### Optická a rentgenová kontrola

Lepší testovací stanice obsahují automatizované systémy založené na optické (*Automated Optical Inspection* (AOI)) či rentgenové (*Automated X-Ray Inspection* (AXI)) kontrole. Jak už z principu vyplývá, v případě AOI a AXI se jedná především o mechanickou kontrolu. Dalším velmi častým způsobem je tzv. „testování uvnitř obvodu“ (*In-Circuit Testing* (ICT)). Jedná se o elektrickou kontrolu testované DPS [21, 5].

Na Obr. 1.3 jsou zobrazeny nejdominantnější se vyskytující poruchy detekované při procesu AOI-AXI-ICT – byly zachovány jejich anglické výrazy kvůli zamezení nesprávného vyložení zapříčiněného překladem. Jak je v diagramu také naznačeno, ICT není schopen detekovat všechny možné funkční poruchy – k tomu bývá určen *Functional Board Test* (FBT) popsany v sekci 1.3.4) [21].



Obr. 1.3: Poruchové módy AOI-AXI-ICT procesu  
(převzato a upraveno z [21])

### 1.3.3 Testování uvnitř obvodu

Již zmíněné ICT spočívá v tom, že testovaná DPS je už při návrhu uzpůsobena tomu, aby se daly „připojit“ potřebné testovací sondy. Hlavním smyslem tohoto testování je ověřit následující:

- Zkontrolovat, zda jsou na konkrétních místech dodrženy doporučené hodnoty napětí podle datových listů osazených komponent.
- Ověřit, že se na DPS nenachází zkratky (nežádoucí spojení vodivých cest během výroby tam, kde by spojení být neměly).
- Ověřit, že DPS neobsahuje otevřené obvody (místo, kde je nechtěné přerušení elektrické cesty – např. špatným osazením a následným připájením součástky).

- Změřit důležité parametry osazených pasivních součástek (elektrický odpor, indukčnost, kapacita) ideálně dříve, než je připojeno napájecí napětí – díky tomu je zaručeno vyšší pravděpodobnosti, že nedojde k poškození testované desky již při „zapnutí“.
- Změřit důležité parametry osazených digitálních a analogových součástek nebo obvodů, které nedosahují velké složitosti (tranzistor, operační zesilovač, integrovaný obvod. . .).
- Kontrola, zda jsou osazeny správné komponenty a nedošlo tak k záměně za jiný (takovou chybu nemusí být snadné najít).

Limitací této metody je především skutečnost, že zdaleka ne všechny uzly obvodu jsou přístupné a nelze je tedy tímto způsobem testovat. Typicky bývá dostupných přibližně 70 % uzlů. Digitální komponenty, především ty komplexní jako procesory a paměti, zpravidla nebývají zcela testovány – definitivně ne touto metodou [21].

### Testovací body a sondy

V případě testovacích bodů (*Test Point* (TP)) se jedná o malé kovové plošky, na jejichž místě dojde ke kontaktu mezi přizpůsobovací (či rovnou testovací) a testovací deskou. Velikost těchto plošek závisí např. na velikosti DPS, přesnosti nástrojů pro výrobu DPS nebo na tvaru a toleranci použitých testovacích sond (*test probes*), které navazují kontakt s TP (např. hojně využívané POGO piny – viz Obr. 1.4).



Obr. 1.4: POGO piny (převzato a upraveno z: <https://ae01.alicdn.com/kf/...>)

Na volbu pozice TP a stejně tak testovacích sond je třeba myslet již při návrhu DPS (viz sekce 2.1.3). Přestože jsou TP většinou umísťovány ze spodní strany desky, rozhodně není považováno za chybný návrh umístění TP na vrchní stranu, avšak není doporučováno kombinování těchto dvou způsobů. Průměr TP je nutné volit dostatečně velký, aby bylo zaručeno, že dojde ke kontaktu s testovací sondou [21].

### 1.3.4 Test funkčnosti DPS

Test funkčnosti DPS (FBT) spočívá v ověření funkčnosti konkrétních částí desky vzhledem k požadovanému chování specifikovaného při návrhu. Schopnost takový test provést závisí na technice důkladném pochopení funkcionality testované DPS, na základě čehož jsou navrženy testovací scénáře (viz kapitola 5) [21].

Hlavní podstata FBT spočívá v nalezení nesprávného chování (poruchy – *failure*) a v identifikaci i specifikaci zdroje této chyby (vady – *fault*). FBT musí být schopen provést nápravu zjištěného poruchového stavu na základě informací o chybě získaných při AOI-AXI-ICT procesu. Realizace FBT je složitá a časově náročná záležitost, nicméně v mnoha případech je vysoce žádoucí takový test používat [21].

## 1.4 Standardy rozhraní pro sériovou komunikaci

V rámci této podkapitoly je popsán komunikační standard TIA/EIA-485 (*Telecommunications Industry Association/Electronic Industries Alliance* (TIA/EIA)), protože je to jeden z požadavků na vlastnosti testovacího zařízení. První podsekcí sekce 1.4.1 se zabývá standardem TIA/EIA-232, jelikož se jedná o předchůdce TIA/EIA-485, a mají tak mnoho společného. Následuje podsekcí, kde jsou zmíněné standardy vzájemně srovnány.

### 1.4.1 TIA/EIA-485 (RS-485)

Standard RS-485 (a také RS-232 i RS-422) popisuje pouze fyzickou vrstvu *International Organization for Standardization/Open Systems Interconnection* (ISO/OSI) modelu a definuje tedy elektrické charakteristiky vysílačů a přijímačů. Pro zajištění schopnosti komunikace je nutné doplnit komunikační protokol, který zastává např. zpracovávání adres zařízení, kontrolní součet, kontrolu kolizí paketů, definuje topologii sítě (např. *Master/Slave*), konstrukci paketů atd. (viz spojová vrstva v Tab. 1.2).

### TIA/EIA-232 (RS-232)

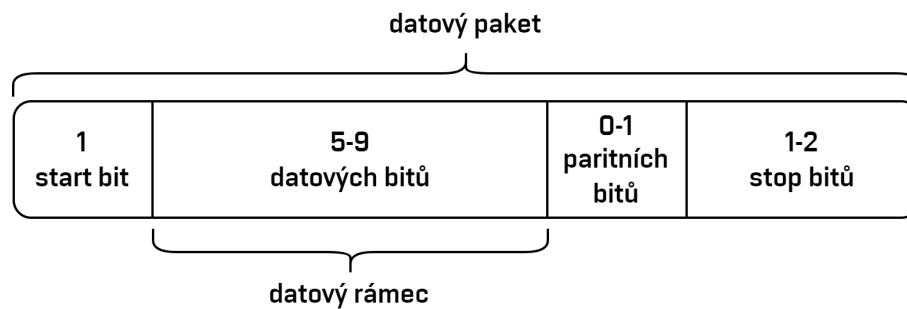
Asynchronní sériový komunikační protokol TIA/EIA-232 nebo také RS-232 (*Recommended Standard* (RS)) je i dnes stále velmi často používán. Zajišťuje komunikaci mezi vysílačem/terminálem (*Data Terminal Equipment* (DTE)) a přijímačem/komunikačním zařízením (*Data Communications Equipment* (DCE)) ve dvoubodovém (*point-to-point*) režimu – lze tedy propojit vzájemně pouze 2 zařízení.

Jako každý komunikační protokol založený na protokolu *Universal Asynchronous Receiver-Transmitter* (UART), i RS-232 vyžaduje synchronizační značky vkládané na začátek a konec datového rámce. Ve výsledku se tedy jedná o datový paket

Tab. 1.2: Vrstvový model Modbus RTU využívající TIA/EIA-485(232) na fyzické vrstvě (Převzato a upraveno z [10])

Vrstva	ISO/OSI funkce	Modbus RTU funkce
7	aplikační	Aplikační protokol Modbus
6	prezentační	
5	relační	↕
4	transportní	
3	síťová	
2	spojová	Modbus sériová linka – Master/Slave
1	fyzická	TIA/EIA-485 nebo TIA/EIA-232

obsahující *start bit* (1 bit), datový rámeček (*data frame*) o 5-9 bitech, paritní bit (žádná/sudá/lichá parita – nepovinný mechanismus zajišťující 1-bitovou kontrolu po přenosu datového paketu) a na závěr *stop bit* (1-2 bity – 2 v případě žádné parity (pevně log. 1 na dané pozici)). Datový paket je znázorněn na Obr. 1.5.



Obr. 1.5: RS-232 – přenášený datový paket

Aby taková asynchronní komunikace fungovala, musí být obě komunikující zařízení souhlasně nakonfigurována, tzn. musí znát správný počet bitů přenášeného datového paketu (pro všechny pakety stejný) a rychlost přenosu (*baud rate*, typicky 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600, 1000000, 1500000).

Z odstavců výše již vyplývá, že přenášená data reprezentují logické hodnoty 0 a 1. Logická 1 (*mark*) odpovídá napětí v rozmezí od -3 V do -15 V a logická 0 (*space*) odpovídá napětí v rozmezí od +3 V do +15 V. Uvedené hodnoty jsou vztaženy k přijímači (DCE), pro vysílač platí spodní hranice -5 V a +5 V. Napěťové úrovně jsou vztaženy vůči jednomu společnému vodiči připojenému k referenčnímu potenciálu označovaném jako země (*ground* (GND)). Toto propojení je nazýváno *single-ended* nebo také *unbalanced* [6].

## Srovnání RS-485 s RS-232

Ještě před samotným srovnáním je vhodné zmínit komunikační standard RS-422, který je RS-485 velmi blízký. Návrh těchto standardů byl zaměřen na optimalizaci maximální délky kabelu a maximální přenosové rychlosti [6].

Zásadní změnu oproti RS-232 představuje přechod na diferenciální (*balanced*) elektrickou signalizaci namísto single-ended (*unbalanced*). To znamená, že spíše než využívání jednoho vodiče vztáženého k zemi (pro jeden směr přenosu) je využíváno vyhrazeného páru vodičů pro každý kanál. Toto řešení dosahuje mnohem vyšší odolnosti vůči indukovanému rušení skrze elektrostatickou vazbu. Šum je totiž naindukován totožně do obou vodičů vyhrazeného páru (tzv. souhlasné (*common-mode*) rušení) a vzhledem ke skutečnosti, že přijímač reaguje na rozdíl napětí mezi vodiči v páru, komunikace nebude tímto rušením ovlivněna [6].

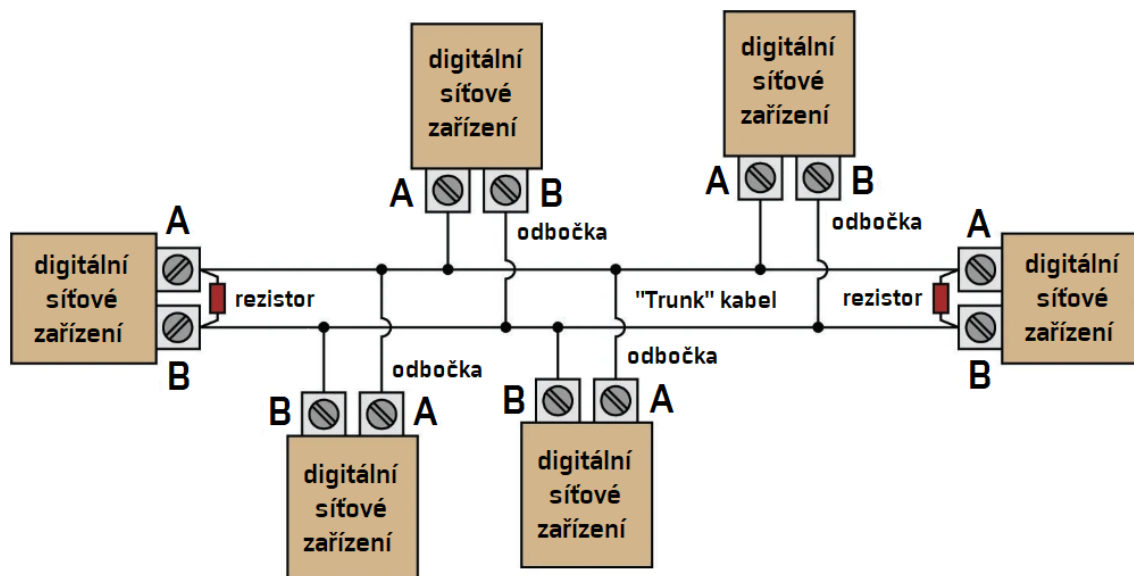
Zatímco RS-422 je tzv. jednocestný (*simplex/one-way*) komunikační standard, RS-485 umožňuje komunikaci oběma směry (*duplex/two-way*) a v obou případech je podporováno více zařízení na jednom segmentu sítě. V případě RS-422 se jedná o 1 DTE zařízení a až 10 DCE zařízení, zatímco RS-485 umožňuje až 32 DTE zařízení v režimu *half-duplex* (zařízení schopná vysílání i přijímání, avšak nikoli naráz) a více DCE zařízení (max. 32). V rámci obou těchto standardů se často hovoří o tzv. *multidrop* síti, tzn. DTE zařízení dokáže posílat zprávu několika DCE zařízením. RS-485 také dokáže propojit 2 zařízení pomocí dvou vyhrazených párů a zprostředkovat tak mezi nimi full-duplex spojení [6].

Vzhledem k potlačení souhlasného šumu při diferenciálním přenosu mohou přijímače rozeznávat logické úrovně již při rozdílu větším než  $\pm 200 \text{ mV}$ , a to je výrazný posun oproti RS-232, kde je minimální rozeznatelná úroveň vůči zemi  $\pm 3 \text{ V}$ . Vysílače protokolu RS-422 musí generovat signály o amplitudě alespoň  $\pm 2 \text{ V}$ , kvůli zajištění minimální šumové rezervy 1,8 V mezi DTE a DCE. Vysílače RS-485 mají povoleno menší šumové rozpětí, tedy minimální úrovně signálu jsou  $\pm 1,5 \text{ V}$  [6].

Největší doporučená délka kabelu pro RS-422 i RS-485 je 1200 m. Maximální přenosová rychlost je totiž nepřímo úměrná délce kabelu, ale oproti RS-232 je podstatně větší díky odolnosti proti šumu již zmiňované diferenciální signalizace. Instalace na dlouhé vzdálenosti (nutnost dlouhých kabelů) nebo aplikace pracující s velkými přenosovými rychlostmi vyžadují tzv. zakončovací rezistory (*terminating resistors*) za účelem eliminace odražených signálů. Tyto odpory by měly být nejvýše dva v síti a to na dvou nejvzdálenějších místech, jež spojuje tzv. kmenový (*trunk*) kabel. Také by měl být brán zřetel na délky kabelů tvořící odbočky (*stub*) z trunk kabelu a zajistit, aby byly co nejkratší (viz Obr. 1.6).

Zaběhlá konvence značení signálů RS-485 v průmyslových zařízeních je „A“ a „B“, alternativně „-“ a „+“ nebo „A-“ a „B+“ [6].





Obr. 1.6: RS-485 – síťové zapojení pro komunikaci v režimu half-duplex  
(Převzato a upraveno z: <https://control.com/textbook/digital-data-acquisition...>)

Tab. 1.3: Srovnávací tabulka elektrických vlastností RS-232, RS-422, RS-485

Vlastnost	Standard		
	RS-232	RS-422	RS-485
Napěťový systém	single-ended	diferenciální	diferenciální
Max. DTE/DCE na linku	1/1	1/10	32/32 <sup>a</sup>
Konfigurace spojení	point-to-point	multidrop	multidrop
Max. provozní vzdálenost	15 m/50 ft	1200 m/4000 ft	1200 m/4000 ft
Max. rychlost přenosu	cca 350 kbit/s	10 Mbit/s	10 Mbit/s <sup>b</sup>
Mód přenosu	full-duplex	simplex	half/full duplex <sup>c</sup>
Rozsah výst. napětí DTE	$\pm 5 V$ až $\pm 15 V$	$\pm 2 V$ až $\pm 5 V$	$\pm 1,5 V$ až $\pm 5 V$
Zakončovací impedance <sup>d</sup>	120 $\Omega$	120 $\Omega$	120 $\Omega$
Rozsah vst. napětí DCE	$\pm 3 V$ až $\pm 15 V$	$\pm 7 V$	$-7 V$ až $+12 V$
Rozlišitelnost DCE	$\pm 3 V$	$\pm 200 mV$	$\pm 200 mV$
Min. vst. impedance DCE	3 až 7 k $\Omega$	4 k $\Omega$	12 k $\Omega$

<sup>a</sup>Pouze jeden DTE aktivní v daný okamžik.

<sup>b</sup>Odpovídá vzdálenosti 10 m/33 ft.

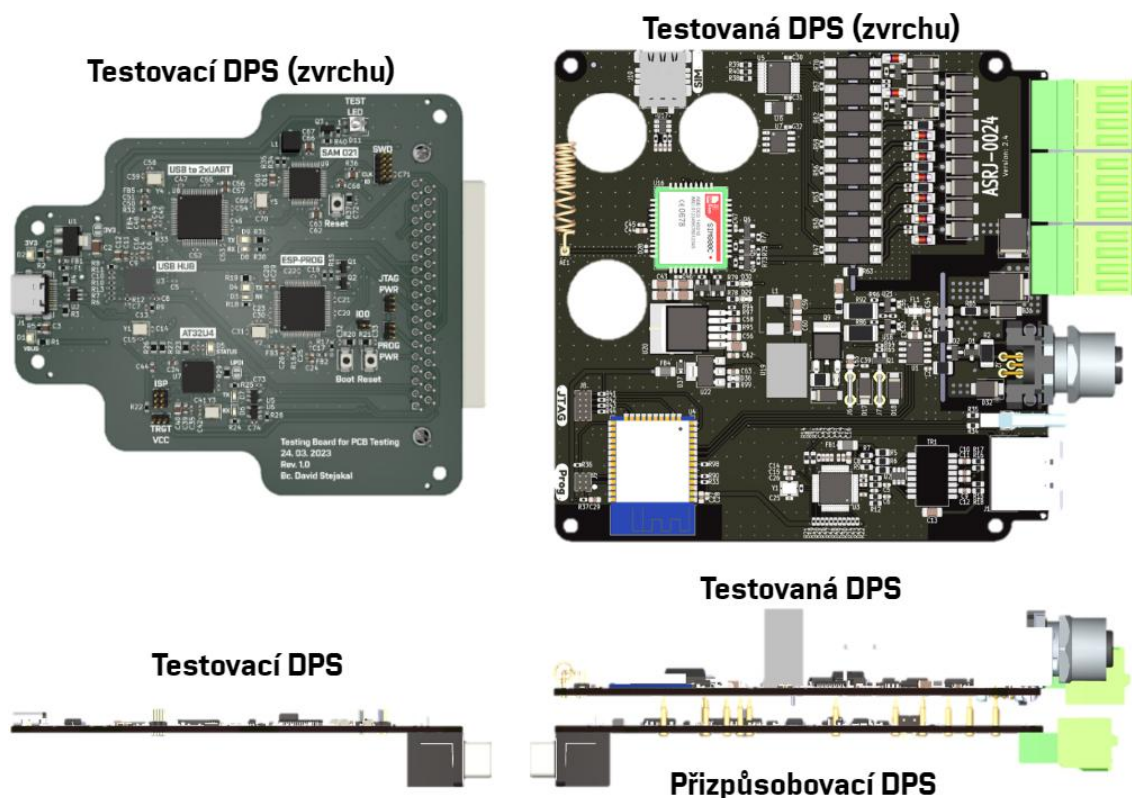
<sup>c</sup>Half-duplex (2 vodiče) a full-duplex (4 vodiče – praktické pouze mezi 1 DTE a 1 DCE).

<sup>d</sup>Hodnota závisí na konkrétní aplikaci, musí odpovídat charakteristické impedanci kabelu.

Popsané (i některé další) parametry komunikačních protokolů RS-232, RS-422 a RS-485 jsou uvedeny v souhrnné Tab. 1.3 [7, 8, 17].

## 2 Návrh a realizace desek

Tato kapitola podrobně pojednává o návrhu a následné realizaci přizpůsobovací a testovací DPS. Pro lepší představu je na Obr. 2.1 zobrazena vzájemná konfigurace mezi těmito deskami. Lze vidět, že testovaná DPS bude umístěna testovacími ploškami na příslušné POGO piny přizpůsobovací DPS. Skrze tyto piny budou přenášena a měřena požadovaná napětí, a také zajistí kontakt pro komunikaci po USB, RS-485 i programovacím rozhraní čipu ESP32.



Obr. 2.1: Vzájemná konfigurace přizpůsobovací a testovací DPS

V případě prvních dvou podkapitol jsou vždy nejdříve uvedeny blokové diagramy navrhovaných desek (Obr. 2.2 a Obr. 2.24). Po nich následují sekce s popisem jednotlivých funkčních bloků blokových diagramů a sekce s popisem konkrétních vybraných komponent. V Tab. 2.2 a Tab. 2.3 jsou uvedeny nejpodstatnější komponenty se stručnými popisy, které byly vybrány k realizaci navrhovaných DPS.

Podkapitola 2.3 se zabývá rozhraním mezi oběma deskami. V Tab. 2.4 je znázorněno podrobné rozdělení jednotlivých signálů 37-pinového konektoru D-Sub DC-37 do konkrétních skupin.

V poslední podkapitole 2.4 je prezentována jak finální verze samostatného celku testovací stanice, tak i sestava přizpůsobovací s testovanou DPS.

K tvorbě schémat elektrických obvodů a také následným návrhům desek plošných spojů včetně 3D modelů byl použit volně dostupný software Kicad 7.0, který je kompatibilní hned s několika platformami. Kicad 7.0 poskytuje také velmi nápomocné nástroje při tvorbě DPS jako například prohlížeč a vykreslovač 3D modelů DPS, různé návrhové kalkulatory, *gerber* prohlížeč pro kontrolu výrobních podkladů nebo dokonce integrovaný SPICE simulátor pro ověření správného chování elektrických obvodů.

## 2.1 Přizpůsobovací DPS

Hlavní podstatou návrhu přizpůsobovací DPS je výhoda, že bude-li v budoucnu změněna testovaná DPS (nová verze, změna rozložení testovacích pinů, výměna MCU za jiný, či zcela jiný produkt), nebude nutné měnit celý testovací systém, ale pouze přizpůsobovací DPS, přičemž testovací DPS bude moci zůstat nedotčena.

Na Obr. 2.2 je ukázán blokový diagram přizpůsobovací DPS. Veškerá rozhraní včetně samotného napájení jsou do DPS přivedena 37-pinovým konektorem DC-37. Pouze napájení dig. vstupů/výstupů (VDC), které může kvůli testovacím účelům nabývat 18 až 40 VDC, je do desky přivedeno externě přes šroubovací svorkovnici. Totéž napětí je zároveň napájecím napětím testované DPS a z toho důvodu je potřeba jej také měřit.

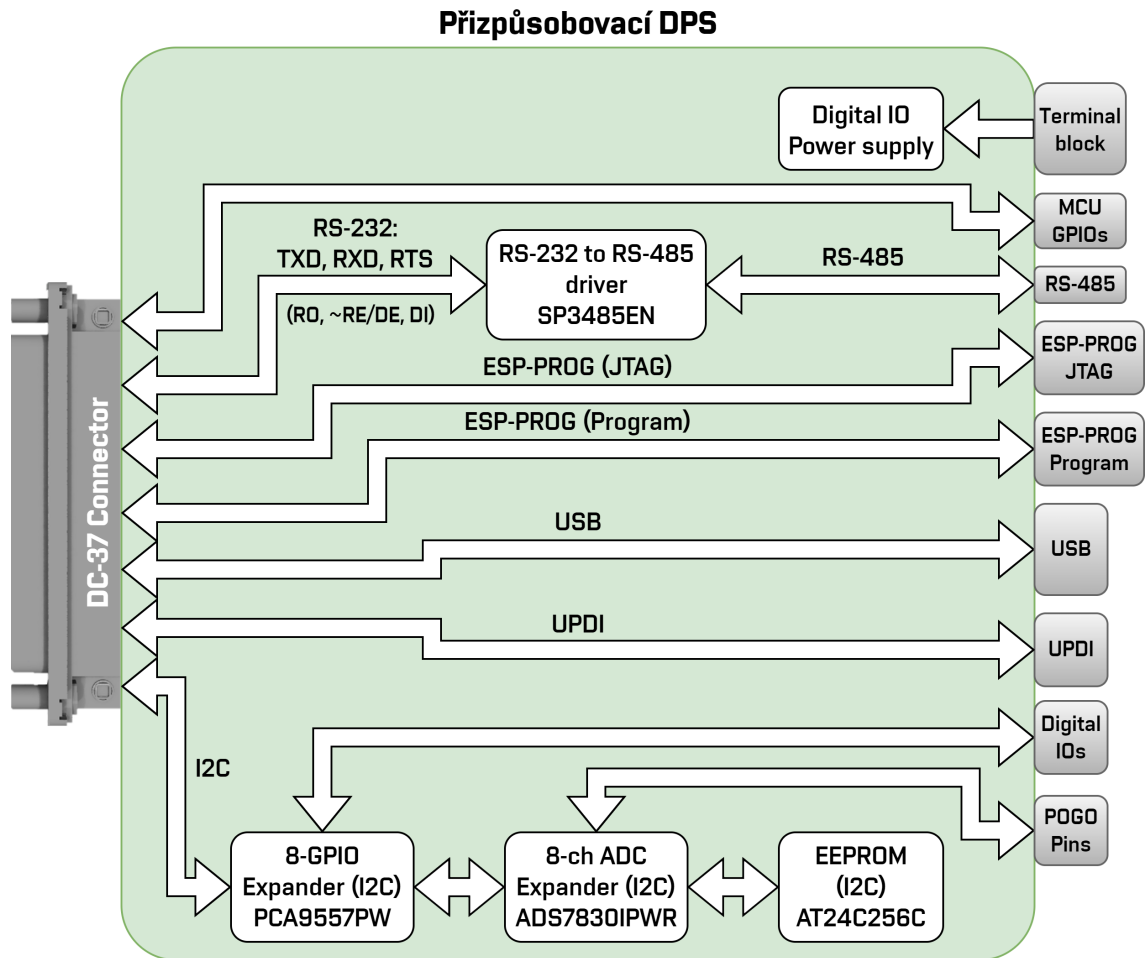
### 2.1.1 Funkční bloky elektrické části návrhu

V následujících podsekcích jsou podrobněji popsány funkční bloky, které byly použity při návrhu elektrického schématu přizpůsobovací DPS. Zmíněnými funkčními bloky jsou: lineární regulátor na  $+3,3\text{ V}$ , převodník z RS-232 na RS-485 s přídavnou ochranou proti *Electrostatic Discharge* (ESD), *Electrical Fast Transient* (EFT) a výboji, oktálový GPIO expandér komunikující po I2C, 8-kanálový A/D (*analog to digital*) převodník komunikující po I2C, 256 kb EEPROM komunikující po I2C.

#### Lineární regulátor napětí

Čip AMS1117-3.3 je typickým lineárním regulátorem napětí využívaným ve spoustě designů díky své jednoduchosti použití, ceně a také svým vlastnostem. Jedná se o typ regulátorů nazývaný *low dropout*, které se vyznačují, jak už z názvu vyplývá, svým nízkým požadovaným minimálním rozdílem mezi vstupním a výstupním napětím (1,1-1,3 V při výstupním proudu maximálně 0,8 A). Obvod je tedy naprosto vyhovující při potřebě převodu napětí z 5 V na 3,3 V.

Při návrhu je nutné brát ohled na maximální výstupní (odebíraný) proud, protože ačkoli je v datovém listu uvedena hodnota 1 A, v poznámkách je uvedeno,



Obr. 2.2: Blokový diagram přizpůsobovací DPS

že maximální výkonové zatížení obvodu činí (pro použité pouzdro SOT-223) pouhý 1,2 W. Tato výkonová ztráta se počítá jako součin rozdílu vstupního a výstupního napětí s výstupním proudem, tedy:  $P_d = (U_{IN} - U_{OUT}) \cdot I_{OUT}$ . Je-li uvažováno vstupní napětí 5 V a výstupní napětí 3,3 V, může výstupní proud dosahovat maximálně 706 mA. Obdobně je v datovém listu zmíněno, že uvedené *dropout* napětí platí pro odebíraný proud do 0,8 A – při vyšším odběru dosahuje i dropout napětí vyšších hodnot.

### Převodník z RS-232 na RS-485

K převodu sériové komunikace RS-232 na diferenciální verzi RS-485 byl využit řadič SP3485EN od výrobce MaxLinear, který poskytuje velmi sofistikované integrované řešení pro tuto úlohu. Společně s RS-485 podporuje také standard RS-422.

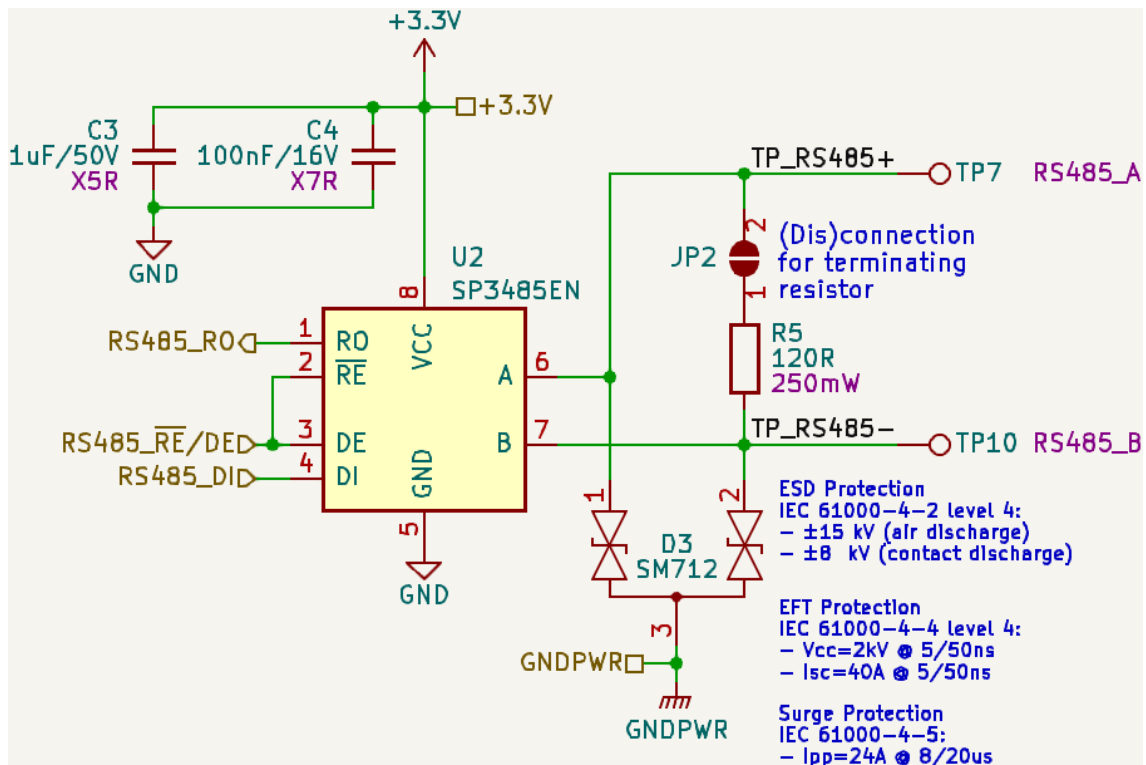
Obvodové zapojení tohoto čipu není nijak složité (viz Obr. 2.3), avšak je na místě zmínit zakončovací impedanci a ochranu sériové komunikační linky.

Jako zakončovací impedance byl zvolen rezistor o standardní hodnotě  $120\ \Omega$  (kvůli možnosti odpojení zakončovacího rezistoru byl přidán pájecí jumper). Při jeho výběru bylo nutné brát ohled také na výkonovou ztrátu na něm – z datového listu lze vyčíst, že rozdílové výstupní napětí při *single-ended* impedanci  $27\ \Omega$  dosahuje hodnot v rozmezí  $1,5$  až  $3,3\ V$ . Pro výpočet výkonové ztráty na rezistoru lze použít notoricky známý vzorec:

$$P_R = \frac{U_{OD\_max}^2}{R}, \quad [W] \quad (2.1)$$

kde symbol  $U_{OD\_max}$  představuje maximální rozdílové výstupní napětí mezi výstupy A, B a symbol  $R$  odpovídá hodnotě zakončovacího rezistoru. Pro zakončovací rezistor na Obr. 2.3 tak platí následující výpočet:

$$P_R = \frac{3,3^2}{120} = 91\ mW.$$



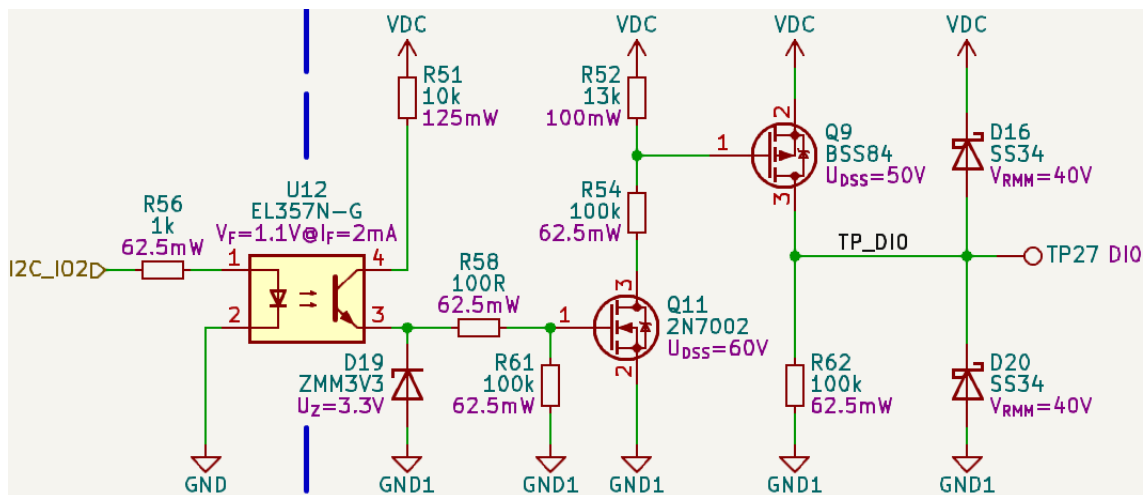
Obr. 2.3: Řadič z RS-232 na RS-485

Ochranu proti elektrostatickému výboji (ESD), rychlým elektrickým přechodovým jevům (EFT) a proti přepětí zajišťuje dodatečný komponent PSM712, který představuje asymetrickou sestavu čtyř *Transient Voltage Suppression diode* (TVS). Jedná se o obvod navržený přímo pro aplikace využívající RS-485. Konkrétní hodnoty odolností tohoto obvodu jsou uvedeny taktéž na Obr. 2.3.

## Oktalový GPIO expandér a výkonové digitální výstupy

V pravém horním rohu přizpůsobovací desky se nachází 2 digitální výstupy a 6 digitálních vstupů (směr je vztažen, možná trochu zmatečně, k testované desce, tedy např. výstup z testované do přizpůsobovací DPS). Standardně je na těchto pinech očekáváno 24 V, avšak kvůli účelům testování na přepětí jsou obvody navrženy tak, aby byly schopny pracovat až do 40 V. Zmíněné digitální výstupy jsou měřeny pomocí 8-mi kanálového ADC (viz sekce 2.1.1). Za účelem ovládání 6-ti digitálních vstupů (či výstupů vzhledem k přizpůsobovací DPS) obsahuje přizpůsobovací DPS oktalový GPIO expandér, který je ovládán po sběrnici I2C z důvodu šetření počtu signálů mezi testovací a přizpůsobovací DPS.

Na Obr. 2.4 lze vidět finální verzi obvodu pro ovládání jednoho digitálního výstupu (vstupu vzhledem k testované desce). Signálem z GPIO expandéru je ovládána infračervená dioda, na jejíž intenzitu reaguje fotocitlivý tranzistor uvnitř optočlenu. Použitím optočlenu je získáno hned několik výhod. První výhodou, a v tomto případě i nejdůležitější, je elektrické oddělení logického obvodu pracujícím na 3,3 VDC od výkonového obvodu operujícím typicky na 24 V (stejnoseměrných či střídavých). Tím je zajištěna odolnost proti napětovým rázům způsobeným například úderem blesku v blízkosti zařízení. Dalším benefitem bývá velmi často dosažení velmi efektivního potlačení elektrického šumu/rušení, které je většinou naakumulováno na referenčním potenciálu (obecně označovaným jako GND).



Obr. 2.4: Obvod pro ovládání 24 V digitálního výstupu

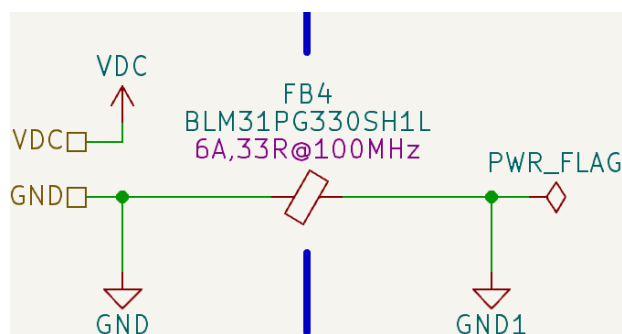
Obvod za optočlenem slouží k výkonovému spínání mezi referenčním potenciálem GND1 a napájecím napětím VDC. K tomu bylo zapotřebí použít 2 *Metal Oxide Semiconductor Field Effect Transistor* (MOSFET) tranzistory – spodní typu N a horní typu P.

Při přivedení vysoké úrovně na diodu optočlenu se otevře tranzistor optočlenu a díky Zenerově diodě (viz D19 na Obr. 2.4) je na vstupu Gate spodního tranzistoru nastaveno její průrazné napětí  $U_F = 3,3 V$ , čímž je tranzistor otevřen ( $U_{GS} = 3,3 V$ ). Sepnutí spodního tranzistoru připne GND1 k odporovému děliči, jehož výstup je připojen na vstup Gate horního tranzistoru. Hodnoty tohoto odporového děliče jsou nastaveny tak, aby při přivedení napájecího napětí VDC v rozmezí 18 V až 40 V bylo zaručeno, že napětí  $U_{GS}$  horního tranzistoru bude alespoň 2 V (horní hranice minimálního prahového napětí  $U_{GS(TH)}$ ) a zároveň nepřekročí maximální hodnotu napětí  $U_{GS(MAX)} = \pm 20 V$  (hodnoty vycházejí z datového listu pro LBSS84LT1G). Tímto je sepnut také horní tranzistor a na jeho výstupu Drain se tak nachází napájecí napětí VDC.

Je-li na vstup optočlenu přivedena nízká úroveň, fotocitlivý tranzistor je uzavřen, a tudíž i spodní tranzistor zůstane nesepnut. Spodní tranzistor se tak chová jako *Open-Drain* a odporovým děličem tedy neteče žádný proud. Horní tranzistor zůstane taky nesepnut a dig. výstup je rezistorem přitažen k zemi (viz R62 na Obr. 2.4).

Rezistor R61 hodnoty 100k připíná vstup Gate k potenciálu GND1 v případě, kdy napětí  $U_{GS}$  spodního tranzistoru nenabývá dostatečné hodnoty pro sepnutí. Stejně tak rezistor R52 hodnoty 13k (vrchní část odporového děliče) připíná vstup Gate horního tranzistoru k napájecímu napětí VDC.

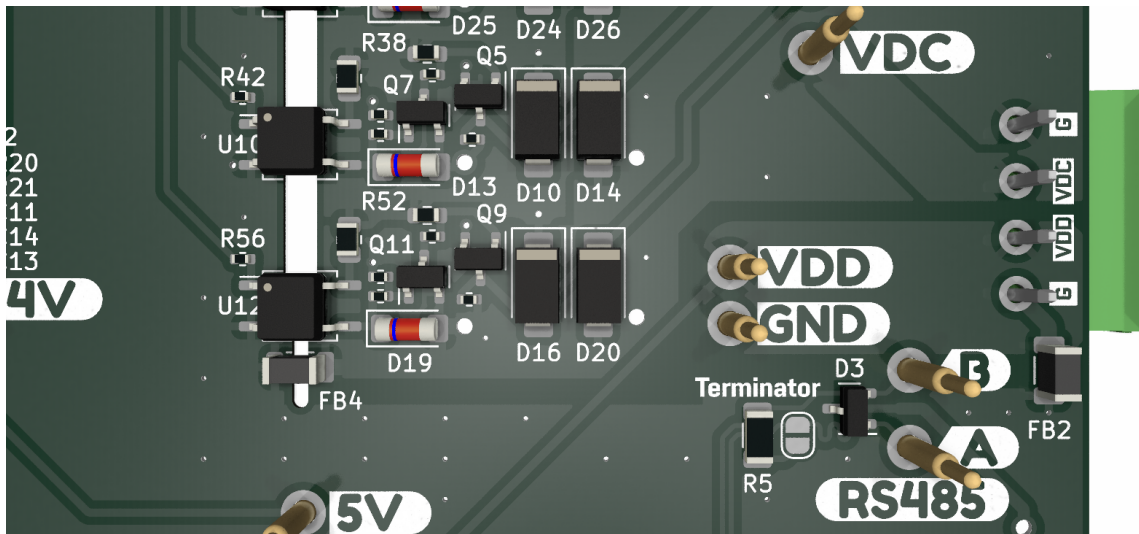
Jak bylo již uvedeno výše, optočlenu zajišťují elektrické oddělení logického obvodu od výkonového. Aby bylo tohoto účinku docíleno, musí být odděleny jednotlivé referenční potenciály, tedy jeden pro infračervené diody (i zbytek log. obvodů) a jeden pro fotocitlivé tranzistory optočlenu (včetně výkonových obvodů). Tyto dva potenciály jsou následně spojeny pouze v jednom bodě (viz Obr. 2.5) skrze výkonovou feritovou perlu (*Ferrite Bead* (FB), více o feritových perlách v sekci 2.1.5). Oddělení referenčních potenciálů je v uvedených schématech znázorněno modrou přerušovanou čarou.



Obr. 2.5: Oddělení referenčních potenciálů logického a výkonového obvodu (FB4)

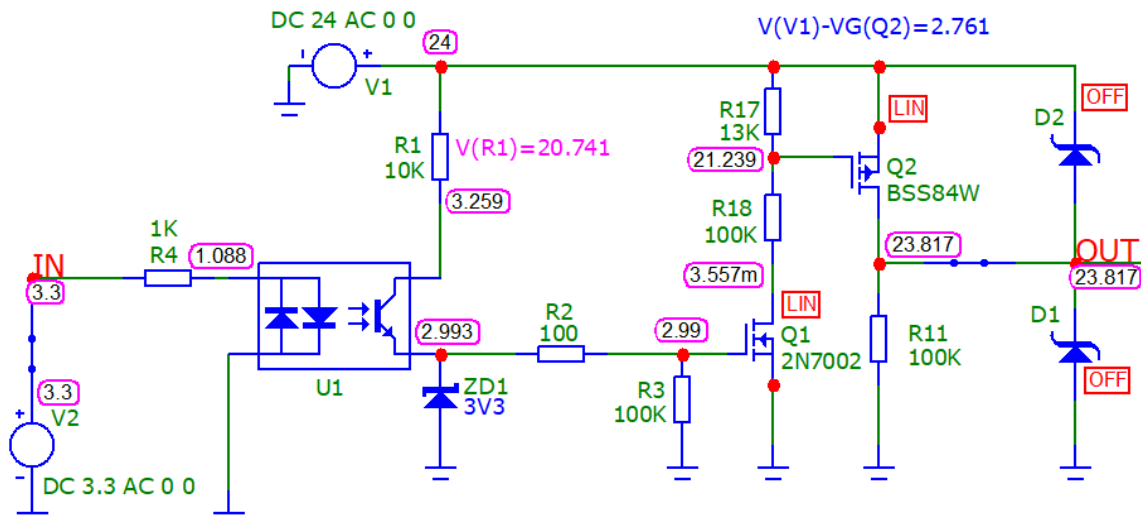
Zmíněnou separaci referenčních potenciálů lze také pozorovat na samotné DPS (viz Obr. 2.6). Zde je patrná také vyfrézovaná mezera pod všemi optočlenu a feri-

tovou perlou, která dodatečně zlepšuje vlastnosti elektrického oddělení vytvořením účinnější bariéry mezi potenciály.



Obr. 2.6: Oddělení referenčních potenciálů logického a výkonového obvodu na DPS

Výše popsaný obvod pro ovládání výkonových digitálních výstupů byl odsimulován v softwaru Micro-Cap 12 od firmy Spectrum Software, jehož licence je od roku 2019 zdarma. V simulaci na Obr. 2.7 lze vidět dříve popsanou funkci, tedy po přivedení vysoké úrovně na vstup optočlenu je otevřen fotocitlivý tranzistor a následně sepnuty oba unipolární tranzistory, čímž je na výstup připnuto napájecí napětí VDC.



Obr. 2.7: Obvod pro ovládání 24 V digitálního výstupu



## 8-kanálový A/D převodník

Ke změření (umožňující následné ověření správnosti) hodnot konkrétních napětí přivedených na testovací piny je využito osmikanálového A/D převodníku komunikujícího po sběrnici I2C. Jelikož měřená napětí mohou nabývat hodnot od 3,3 V do 40 V, bylo nutné tomu přizpůsobit vstupy ADC. Kvůli zvětšení měřicího rozsahu, aby bylo možné měřit napětí přibližně do 5 V bez použití dodatečných obvodových zapojení, byla pro ADC zvolena hodnota napájecího napětí 5 V. Vstupy, na které se očekává přivedení napětí většího než 5 V, musely být opatřeny odporovým děličem, který nastavuje převodový poměr mezi měřeným a kvantizovaným napětím. Hodnoty rezistorů byly voleny s ohledem na co nejnížší chybu převodového koeficientu (Rovnice 2.2), co nejnížší výstupní odpor děliče (Rovnice 2.3) a dostatečně malý stálý proud děličem (Rovnice 2.4).

$$div\_err = \left( \frac{\frac{R_2}{R_1+R_2} \cdot U_{in\_max}}{U_{out\_max}} - 1 \right) \cdot 100 \quad [\%] \quad (2.2)$$

Symbol  $R_1$  představuje hodnotu horního rezistoru a  $R_2$  hodnotu spodního rezistoru odporového děliče,  $U_{in\_max}$  je nejvyšší očekávaná hodnota napětí přivedena na vstup děliče a  $U_{out\_max}$  je maximální požadovaná hodnota napětí na výstupu děliče resp. horní hranice rozsahu vstupu ADC.

Tedy pro odporový dělič na Obr. 2.8 platí následující výpočet:

$$div\_err = \left( \frac{\frac{4300}{30000+4300} \cdot 40}{5} - 1 \right) \cdot 100 = 0,29 \%$$

$$div\_R_{out} = \left( \frac{1}{R_1} + \frac{1}{R_2} \right)^{-1} \quad [\Omega] \quad (2.3)$$

Pro symboly  $R_1$  a  $R_2$  platí stejný význam jako v předchozí rovnici. Odporovému děliči na Obr. 2.8 tak odpovídá následující výpočet:

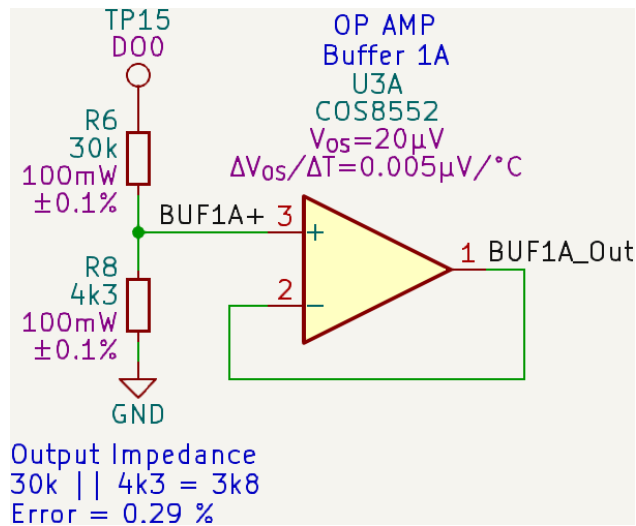
$$div\_R_{out} = \left( \frac{1}{30000} + \frac{1}{4300} \right)^{-1} \approx 3,8 \text{ k}\Omega.$$

$$div\_I = \frac{U_{in\_max}}{R_1 + R_2} \quad [A] \quad (2.4)$$

Symboly  $R_1$  a  $R_2$  mají opět stejný význam jako v předešlých rovnicích a platí tak pro ně následující výpočet:

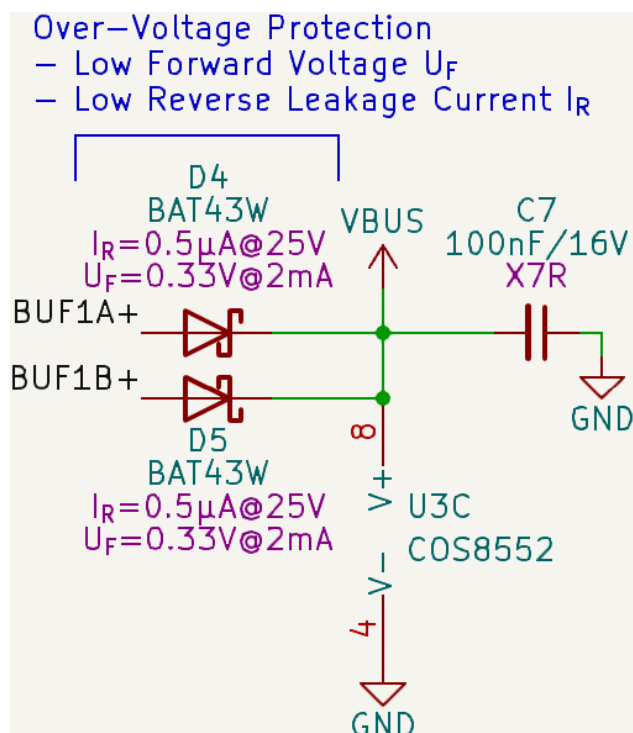
$$div\_I = \frac{40}{30000 + 4300} = 1,17 \text{ mA}.$$

Aby zmíněné odporové děliče nebyly výrazně proudově zatěžovány, čímž by vznikala nezanedbatelná chyba dělicího poměru, byly na jejich výstupy přidány posilovače (*buffer*) z operačních zesilovačů s jednotkovým zesílením (viz Obr. 2.8).



Obr. 2.8: Operační zesilovač s jednotkovým zesílením (U3A) na vstupu ADC

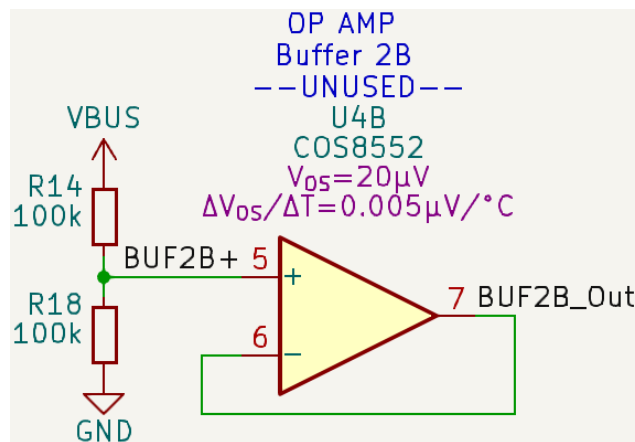
Na vstupy operačních zesilovačů byly přidány Schottkyho diody s nízkým prahovým napětím ( $U_F = 0,33 V$ ) vůči napájecímu napětí (viz Obr. 2.9). Tím je zajištěna ochrana vstupů operačních zesilovačů proti přepětí většímu než  $V_{BUS} + 0,33 V$ .



Obr. 2.9: Ochrana vstupů operačního zesilovače proti přepětí (D4 a D5)

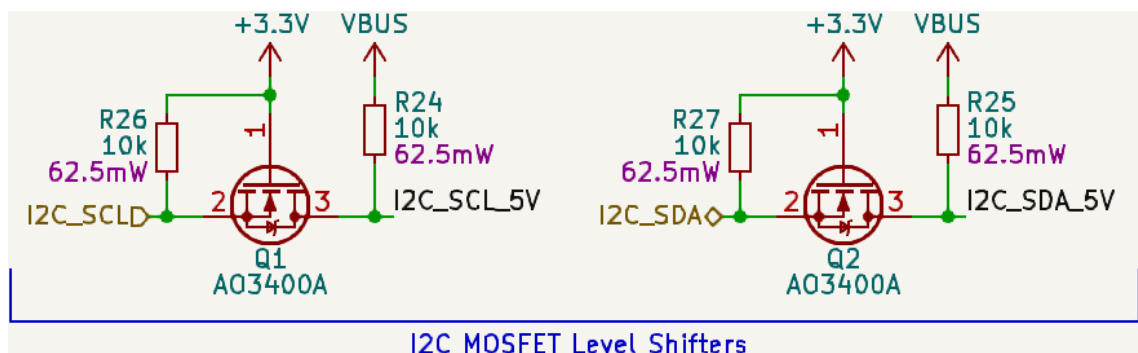
V rámci jednoho pouzdra elektronické součástky COS8552 se nachází 2 operační zesilovače. Jeden nevyužitý operační zesilovač byl zapojen dle Obr. 2.10, aby bylo

zamezeno nežádoucím nahodilému dosažení saturačního stavu, při kterém by vykazoval větší odběr elektrické energie nebo nežádoucím způsobem ovlivňoval přesnost z páru využitého operačního zesilovače [11].



Obr. 2.10: Zapojení nevyužitého operačního zesilovače (U4B)

Kvůli zajištění kompatibility při komunikaci po sběrnici I2C mezi zařízeními napájenými 3,3 V i 5 V bylo nutné přidat dva tzv. *level-shiftery*. Každý je tvořen jedním MOSFET tranzistorem a dvěma *pull-up* rezistory, jak lze vidět na Obr. 2.11.



Obr. 2.11: Obousměrný napěťový převodník mezi 3,3 V a 5 V využívající MOSFET

Funkce převodníku napětí využívající MOSFET spočívá v tom, že je-li na Source tranzistoru přivedena nízká úroveň, tranzistor je otevřen (napětí  $U_{GS} = 3,3 V$ ) a Drain je tedy také přitážen k nízké úrovni. Naopak je-li na Source přivedena vysoká úroveň, tranzistor zůstane uzavřen (napětí  $U_{GS} \approx 0 V$ ) a Drain je přitážen pull-up rezistorem k napětí 5 V. Při opačném směru, tedy když levá strana zastupuje funkci „přijímače“/vstupu, je požadovaná funkce převodníku zajištěna pomocí vnitřní diody tranzistoru mezi vývody Source a Drain – když je pravá strana přitážena k nízké úrovni, levá strana je skrze diodu přitážena také. V opačném případě je pravá strana ve stavu *high* a levou stranu převodníku přitahuje pull-up rezistor k napětí 3,3 V. Tímto je zajištěna funkčnost obousměrného převodníku napětí [4].

## EEPROM

Paměť EEPROM AT24C256C od společnosti Microchip je vnitřně konfigurována do osmi bloků ( $32768 \times 8 = 256$  kb), čemuž odpovídá také možnost adresace zařízení pomocí tří volitelných pinů A2, A1 a A0. Na základě nastavení těchto pinů (připojení přímo k napájení či GND) je zvolen konkrétní paměťový prostor (blok) o velikosti 32768 bitů. Adresovací piny EEPROM jsou na přizpůsobovací DPS pevně připojeny ke GND a byl tak vybrán hned první paměťový blok.

Adresový prostor celého čipu je dále členěn celkem do 512 stran, z čehož vyplývá, že každému bloku odpovídá 64 stran. Každá strana obsahuje 64 bytů paměti, tzn. k celkové velikosti jednoho bloku (uvedené výše) se lze dopočítat následujícím způsobem:  $64 \times 64 \times 8 = 32768$  bitů. Adresace při náhodném zápisu či čtení lze dosáhnout použitím 15-ti adresových bitů:  $\log_2(512 \times 64)$ .

Paměť je v rámci testovací stanice využita k elektronickému záznamu informací týkajících se přizpůsobovací DPS. Na jejich základě by mohl být zvolen testovací scénář (viz kapitola 5) a následně vygenerována výsledná zpráva automatického testu. Konkrétně se jedná o následující informace: název DPS, jméno autora, verze DPS, datum vytvoření konfigurace, stručný popis DPS, konfigurace DPS.

K jednoznačnému určení, kde přesně se uvedená data v paměti nachází, slouží přehled registrů viz Tab. 2.1. Každý registr má v tabulce svou počáteční adresu, jméno, sloupec pro určení konkrétní sekce a sloupec pro rozdělení sekcí na jednotlivé bity dle funkce.

Tab. 2.1: Přehled registrů EEPROM

adresa	jméno	sekce	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x0000	DPSNAME	511:0	DPSNAME[511:0]							
0x0040	AUTHOR	255:0	AUTHOR[255:0]							
0x0060	DPSVER	31:0	DPSVER[31:0]							
0x0064	CFGDATE	79:0	CFGDATE[79:0]							
0x006E	DPSDESC	1023:0	DPSDESC[1023:0]							
0x00EE	DPSCFG	7:0	ADCCHNLS[3:0]			IOEXPCHNLS[3:0]				
		15:8	DICOOUNT[3:0]			DOCOUNT[3:0]				
		23:16			RS485	RS232	CAN	SPI	I2C	USART
		31:24								
		39:32								
		47:40								
0x00F4	Reserved									
...										
0x0FFF										

**Jméno DPS** DPSNAME 0x0000

Registr obsahující jméno desky plošných spojů. Maximální délka je 64 znaků.

**Jméno autora** AUTHOR 0x0040

Registr obsahující jméno autora. Maximální délka je 32 znaků.

**Verze desky** DPSVER 0x0060

Registr obsahující verzi desky plošných spojů. Maximální délka jsou 4 znaky.

**Datum konfigurace** CFGDATE 0x0064

Registr obsahující datum konfigurace desky plošných spojů. Max. délka je 10 znaků.

**Popis DPS** DPSDESC 0x006E

Registr obsahující stručný popis desky plošných spojů. Max. délka je 128 znaků.

**Konfigurace DPS** DPSCFG 0x00EE

Registr obsahující konfiguraci přizpůsobovací desky plošných spojů.

**Bity 21:16** – RS485, RS232, CAN, SPI, I2C, USART Použitá rozhraní  
Nastavený bit znamená výskyt konkrétního rozhraní na přizpůsobovací DPS.

**Bity 15:12** – DICOUNT[3:0] Počet digitálních vstupů  
Hodnota zapsaná do této oblasti definuje počet digitálních vstupů vyskytujících se na přizpůsobovací DPS (resp. 0-15).

**Bity 11:8** – DOCOUNT[3:0] Počet digitálních výstupů  
Hodnota zapsaná do této oblasti definuje počet digitálních výstupů vyskytujících se na přizpůsobovací DPS (resp. 0-15).

**Bity 7:4** – ADCCHNLS[3:0] Počet kanálů A/D převodníku  
Hodnota zapsaná do této oblasti definuje počet kanálů A/D převodníku vyskytujícím se na přizpůsobovací DPS (resp. 0-15).

**Bity 3:0** – IOEXPCHNLS[3:0] Počet vstupů/výstupů GPIO expandéru  
Hodnota zapsaná do této oblasti definuje počet vstupů/výstupů GPIO expandéru vyskytujícím se na přizpůsobovací DPS (resp. 0-15).

**Poznámka:** bity 47:24 jsou rezervovány pro případné rozšíření informací obsažených v konfiguračním registru.

**Nevyužitý prostor** Reserved 0x00F4

Nevyužitý paměťový prostor rezervovaný do budoucna.

## 2.1.2 Popis funkce a vlastností vybraných komponent

V Tab. 2.2 je uveden výběr důležitých komponent použitých při návrhu přizpůsobovací DPS. Ke každé komponentě je v druhém sloupci tabulky uveden také stručný popis a souhrn parametrů.

Tab. 2.2: Podstatné vybrané komponenty na přizpůsobovací DPS

Komponent	Stručný popis
SP3485EN	RS-485 řadič – 1/1 (řadičů/přijímačů), rychlost přenosu se zatěží až 10 Mbps, napájecí napětí 3,3 V, half-duplex
PCA9557PW,118	8×GPIO expandér – podporuje komunikaci po I2C, napájecí napětí 2,3 až 5,5 V, filtr šumu na SCL/SDA vstupech, operační frekvence 0 Hz až 400 kHz, HiZ a Open-Drain na IO0, I/O konfigurační registr
ADS7830IPWR	8-kanálový ADC – podporuje komunikaci po I2C, operační frekvence 100/400/1700 kHz, vzorkovací frekvence 70 kHz, 8 Single-Ended/4 diferenční vstupy, interní napěťová reference 2,5 V, max. $\pm 0,5$ LSB INL/DNL, napájecí napětí 2,7 až 5 V
COS8552	duální operační zesilovač – Rail-to-Rail rozsah vstupů i výstupu, téměř nulový drift nuly ( $0,005 \mu V/^{\circ}C$ ), <i>Gain Bandwidth Product</i> typicky 1,5 MHz, napájecí napětí 1,8 až 5 V
AT24C256C	Paměť EEPROM – komunikace po I2C, napájení 1,7-5,5 V, konfigurace 256K ( $32768 \times 8$ ), organizována do 512 stran po 64 bytech, náhodný zápis/čtení vyžaduje 15-bitovou datovou adresu ( $\log_2(512 \times 64)$ ), umožňuje i 64-bytový zápis (celá strana), Wire-Protect pin
POGO piny	Testovací kolíky – délka 9 mm, hloubka stlačení 2 mm, průměr těla 1,5 mm, průměr základny 2 mm (viz Obr. 1.4)

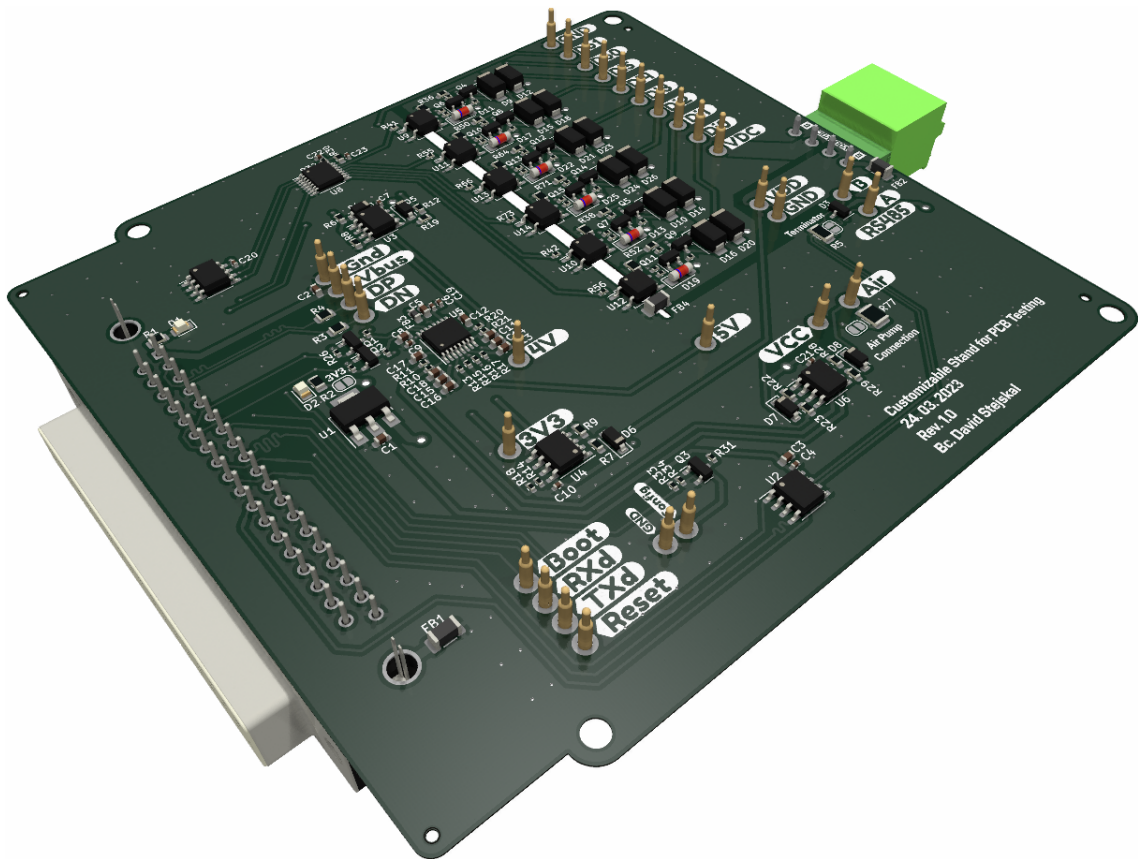
## 2.1.3 Pravidla pro použití TP při návrhu DPS

Jak bylo již popsáno v sekci 1.3.3, při použití TP v návrhu DPS je důležité dbát jistých pravidel. Aby bylo zaručeno, že dojde ke kontaktu mezi TP a testovací sondou, je doporučeno používat průměr TP alespoň 0,035“ (0,89 mm). S volbou průměru může souviset také minimální vzdálenost mezi středy dvou TP, pro níž je doporučované minimum 0,05“ (1,27 mm).

Doporučená vzdálenost TP od okraje desky je  $\geq 0,125''$  (3,18 mm), a to z toho důvodu, že při výrobě desek jsou využívány různé druhy uchopení desek automatizovanými prostředky.

### 2.1.4 Finální výstup návrhu přizpůsobovací desky

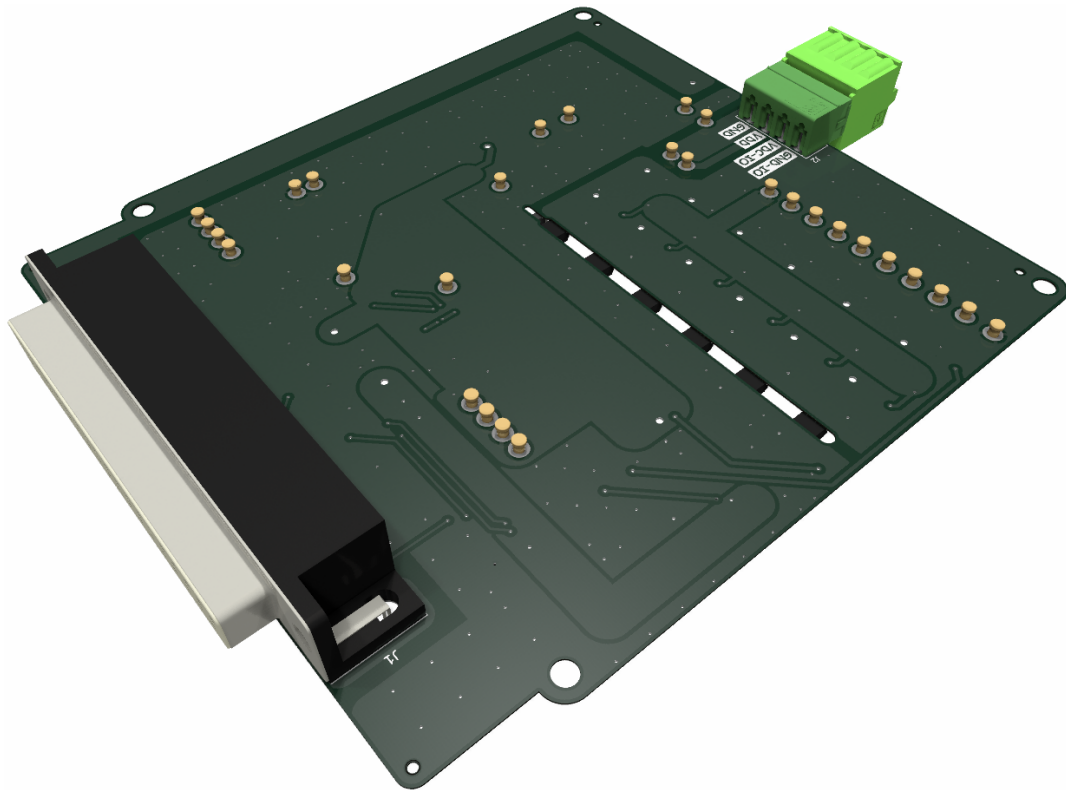
Současně s návrhem přizpůsobovací DPS byl udržován také její 3D model. K jeho zobrazení a následnému exportu byl využit vestavěný nástroj 3D Viewer softwaru Kicad 7.0. Zmiňované 3D pohledy pod úhlem lze pozorovat na Obr. 2.12 a Obr. 2.13.



Obr. 2.12: 3D model přizpůsobovací DPS – pohled shora po úhlem

Na Obr. 2.14 je vygenerovaný výkres měděných ploch vrchní vrstvy přizpůsobovací DPS včetně jejích okrajů a popisků (*silkscreen* vrstva). Na tomto výkresu si lze dobře všimnout například oddělení referenčních potenciálů logické části (GND) a výkonové části (GND1).

Je možné vidět také využití vedení diferenciálních párů, jejichž zakončovací (*single-ended*) impedance  $50 \Omega$  i vzájemná (*differential*) impedance  $90 \Omega$  byly při návrhu řízeny – nejprve bylo nutné zjistit požadovanou tloušťku cesty pro zajištění



Obr. 2.13: 3D model přizpůsobovací DPS – pohled zespodu po úhlem

zakončovací impedance a následně ladit mezeru mezi vodiči diferenciálního páru. K výpočtům byl využit kalkulátor z webu výrobce desek plošných spojů JLCPCB<sup>1</sup>.

Aby byly zajištěny stejné délky obou vodičů diferenciálních párů nebo vysílacího (TX) a přijímacího (RX) vodiče v případě sériové komunikace, bylo při vedení měděných cest využito funkce pro tvorbu meandrů.

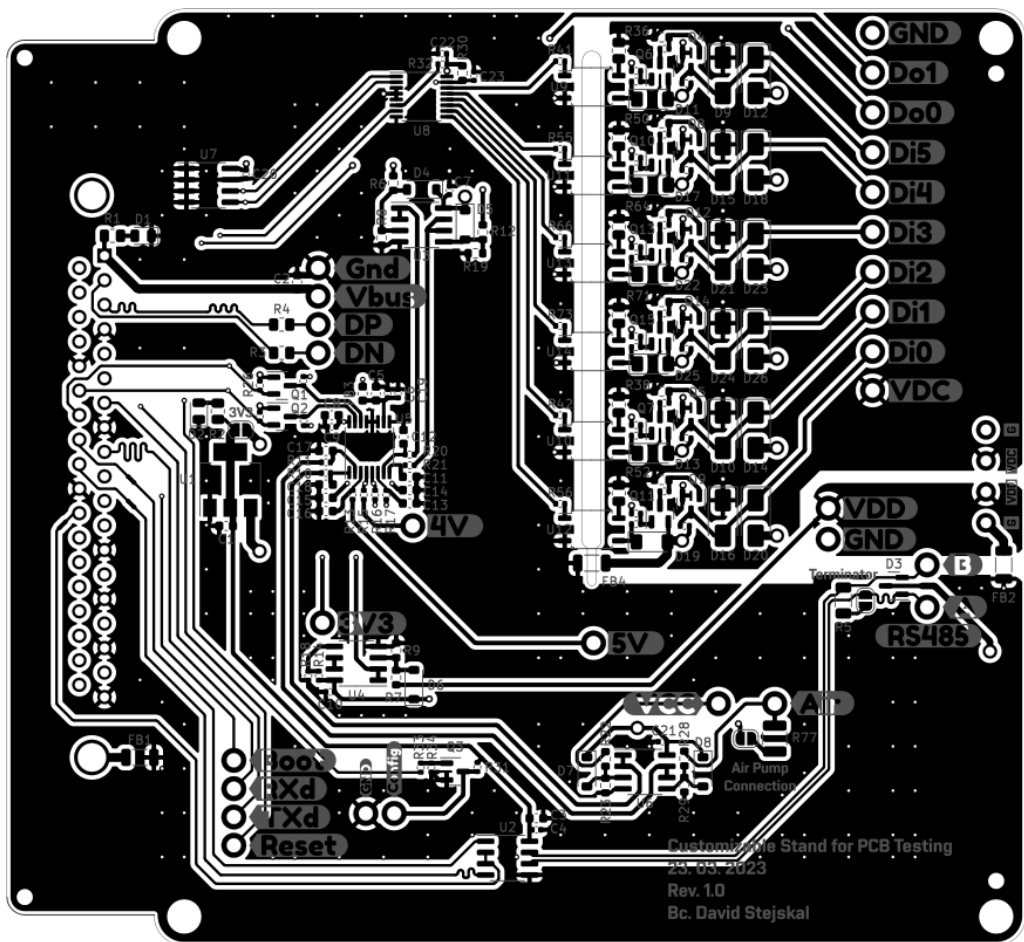
Výrobní výkres s umístěním komponent včetně pájecí masky byl umístěn do příloh (viz Příloha C.1). Na něm lze vidět, že v případě ochrany byl kladen důraz na vhodnost jejich umístění.

Pro ochranu sériové komunikace RS-485 byl využit čip SP3485EN (podrobněji viz sekce 2.1.1), který je umístěn v bezprostřední blízkosti k POGO pinům zprostředkávající požadovaný kontakt (viz Obr. 2.15), aby v případě přítomnosti nežádoucích vlivů bylo vše potlačeno hned při vstupu do desky.

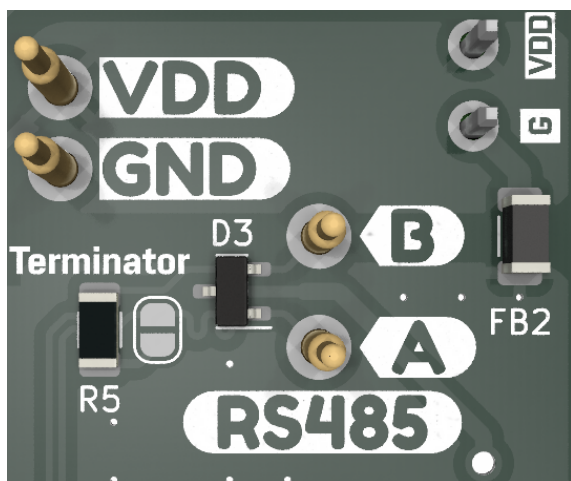
Digitální výstupy v pravém horním rohu slouží ke spínání standardně 24 V, avšak při testování musí odolat napětí až 40 V. Překročení tohoto napětí je zamezeno Schottkyho diodami umístěnými, vzhledem ke zbytku příslušných obvodů, nejbližší odpovídajícím POGO pinům. Příklad zapojení těchto diod v elektrickém obvodu je na Obr. 2.4 (D16, D20). Uspořádání na DPS lze vidět na Obr. 2.16 (D21, D23).

<sup>1</sup>Dostupný z: <https://jlcpcb.com/pcb-impedance-calculator/>

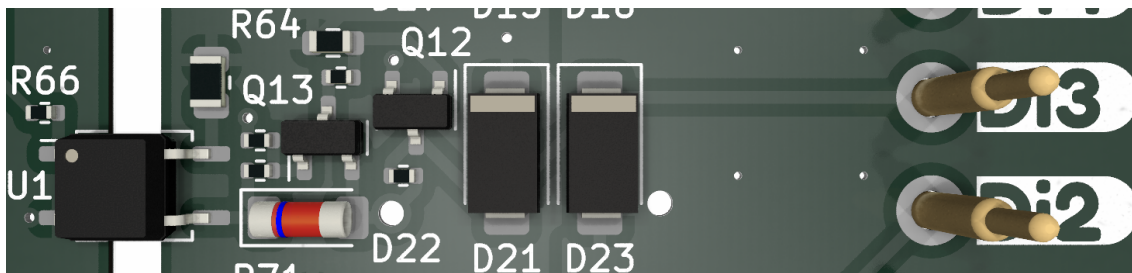




Obr. 2.14: Vrchní vrstva přizpůsobovací DPS – měděná vrstva s potiskem



Obr. 2.15: Ochrana sériové komunikace RS-485 (D3)



Obr. 2.16: Přepětová ochrana digitálních výstupů (D21, D23)

## 2.1.5 Nevyužité části ve finálním návrhu

Jako při každém vývoji, spousta toho, co je navrženo, se nakonec nerealizuje. Tato sekce pojednává o průběhu návrhu některých částí, které ve výsledku nebyly implementovány. Většina se týkala obvodu napájení.

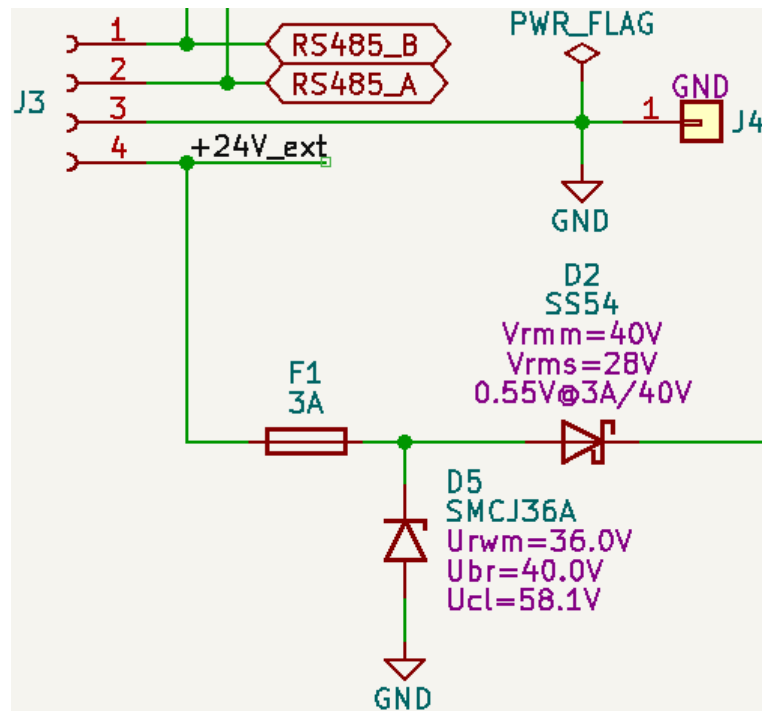
Prvotní myšlenka byla taková, že přizpůsobovací deska bude napájena stejným napětím jako deska testovaná – tedy napětím, jímž bude testována odolnost na podpětí i přepětí. Z toho důvodu byl navržen obvod obsahující přepětovou ochranu, ochranu proti otočení polaroty napájení, filtrace, obvod stabilizace napětí a spínaný DC-DC měnič na 9 V s *Electromagnetic Compatibility* (EMC) filtrem. Zmíněné části jsou popsány v následujících podsekcích.

Hlavním důvodem vynechání uvedených částí bylo zjednodušení celkového návrhu přizpůsobovací desky. Všechny tyto části jsou totiž již součástí testované desky a pro samotnou funkci přizpůsobovací DPS postačí přivedení již stabilizovaných 5 V. Vzhledem k tomu, že v rozhraní vedoucím z testující DPS je zahrnuto také napájení z USB, s výhodou je ho využito.

### Přepětová ochrana

Odolnost DPS proti přepětí byla zajištěna tzv. transilem (TVS). Vybraný TVS zaručuje ESD ochranu úrovně 4 podle normy IEC 61000-4-2 (*International Electrotechnical Commission* (IEC)), která definuje odolnost 15 kV proti výboji vzduchem a 8 kV proti výboji dotykem [19].

Hodnota napětí zlomu (*breaking voltage*) byla zvolena na 40 V, tedy do tohoto napětí nemá TVS žádný zásadní vliv na funkci obvodu. Avšak po překročení této hodnoty začne TVS velmi intenzivně vést proud, čímž vzniká zkrat. Z tohoto důvodu musí být před TVS umístěna tavná pojistka, která se při zkratu (případně při nadměrném odběru proudu) přetaví. Zmíněné bezpečnostní prvky byly umístěny ihned za přívodní napájecí svorku, jak je ukázáno na Obr. 2.17 (pojistka F1 a transil D5).



Obr. 2.17: ESD ochrana – tavná pojistka (F1) a TVS (D5)

### Ochrana proti otočení polarity napájení

Při volbě ochrany DPS proti připojení opačné polarity napájení bylo vybíráno mezi čtyřmi způsoby. První dva využívají buď běžné usměrňovací diody nebo Shottkyho diody. Výhodou těchto dvou variant je jednoduchost a nízká cena. Další dva způsoby využívají MOSFET, přičemž typ s P kanálem umožňuje jednodušší spínání, zatímco typ s N kanálem je schopen pracovat při vyšších proudech díky svému nižšímu odporu při sepnutí označovaném  $R_{ds(on)}$ , ale za cenu potřeby obvodu řízení spínání řídicího vstupu (*gate*).

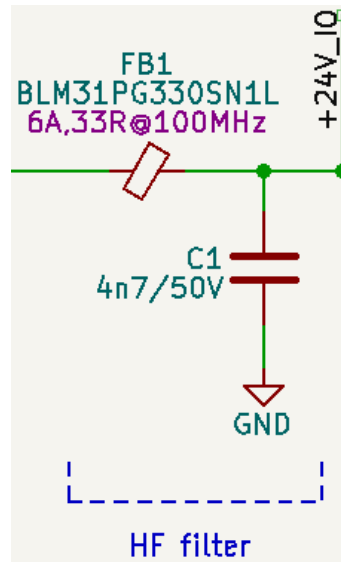
Jelikož pro tuto aplikaci nejsou vysoké nároky na odebíraný proud ( $I_{load}$ ), byl výběr zúžen na první dvě varianty. Rozhodujícím parametrem byl ve výsledku ztrátový výkon ( $P_d$ ) na usměrňovacím členu. Shottkyho diody mají zpravidla nižší prahové napětí ( $U_F(T)$ ) a díky tomu na nich dochází k nižším tepelným ztrátám, které odpovídají vztahu:  $P_d = I_{load} \cdot U_F(T)$ . Zapojení lze vidět na Obr. 2.17 (dioda D2).

### Filtrace externího napájení

V běžné praxi je vyžadováno dobré filtrování šumu napájecího zdroje a také omezení vysokofrekvenčního přeslechu mezi analogovými a digitálními doménami. Pro tuto aplikaci jsou vhodné tzv. feritové perly (FB) [22].

Feritová perla je pasivní zařízení, které filtruje vysokofrekvenční šumovou energii v širokém frekvenčním rozsahu. V zamýšleném frekvenčním rozsahu se chová jako

rezistor a energii těchto vyšších frekvencí (nežádoucí rušení) rozptýluje ve formě tepla. Feritová perla se zapojuje do série s napájecím (či signálovým) vodičem (jak je ukázáno na Obr. 2.18) a bývá často kombinována s kondenzátory zapojenými mezi vodičem a zemí na obou stranách FB. To vytváří nízkofrekvenční filtrační síť, která dále snižuje vysokofrekvenční šum napájecího zdroje [3].



Obr. 2.18: Vysokofrekvenční filtr s feritovou perlou

Účinnost schopnosti filtrovat vysoké frekvence v rezistivním regionu FB velmi rapidně klesá, protéká-li vodičem stejnosměrný proud (*Direct Current (DC) bias*). To je způsobeno nasycením jádra FB. V případě DC napájecího vodiče je předpokládáno, že DC proud vodičem vždy poteče, a proto je s tímto poklesem při návrhu vysokofrekvenčního filtru počítat. Aby bylo dosaženo přijatelné eliminace saturace jádra, doporučuje se volit FB s nominálním proudem alespoň dvojnásobným, než je předpokládaný proud protékající napájecím vodičem [1].

### Stabilizátor napětí

Pro stabilizaci vstupního napětí byl vybrán sériový tranzistorový regulátor se ZD (*Zenerova dioda (ZD)*). Tranzistor je využíván jako emitorový sledovač, jehož napěťové zesílení je velmi blízké jedné, a tedy referenční napětí vytvořené ZD na bázi tranzistoru je udržováno také na výstupu – emitor tranzistoru. Tranzistor v zapojení jako emitorový sledovač funguje jako proudový zesilovač a ZD tak není tolik výkonově namáhána [18].

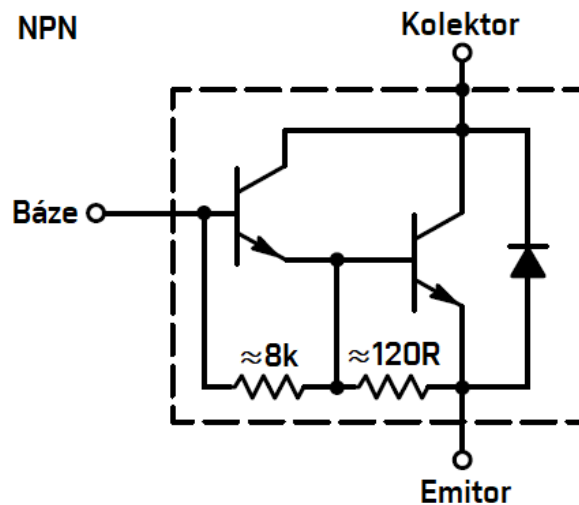
Přidáním oddělovacího/filtračního kapacitoru do báze tranzistoru je „vyhlazeno“ vstupní napětí. Díky tomu, že je kapacitor umístěn na vstup, může pracovat s mnohem nižším proudem (při stejném napětí), a tedy nemusí dosahovat tak velkých

hodnot kapacity, jako by tomu bylo na výstupu. Jedná se o tzv. násobič kapacity z pohledu filtrace zvlnění – jako by byl na výstupu kapacitor hodnoty  $\beta \cdot C$ , kde  $\beta$  ( $h_{21e}$ ) je proudové zesílení tranzistoru a  $C$  kapacita kapacitoru).

Vlastnosti tohoto zapojení jsou:

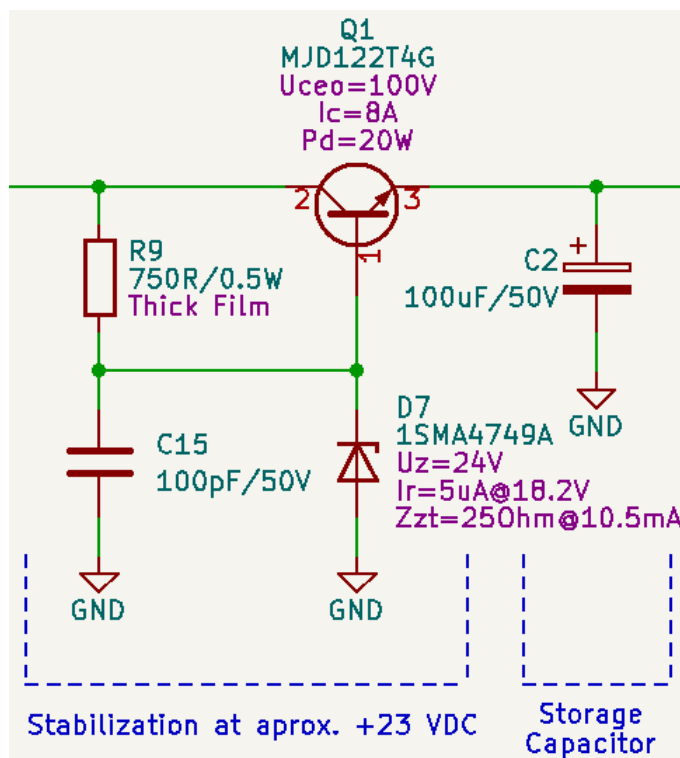
- dobrá efektivita – malé výkonové/tepelné ztráty,
- velké proudové zesílení – velký výstupní (emitorový) proud při malém vstupním (bázovém) proudu,
- malý výstupní odpor,
- výstupní napětí  $U_2 = U_{ZD} - U_{BE}$  ( $U_{ZD}$  – referenční napětí ZD,  $U_{BE}$  – napětí mezi bází a emitorem tranzistoru).

S výhodou získání několikanásobně většího výstupního proudu lze využít zapojení sériového tranzistorového regulátoru s Darlingtonovým párem, jehož proudové zesílení je přibližně součin proudových zesilovacích činitelů tranzistorů tvořících pár. Toto zapojení se může z vnějšku jevit jako jeden tranzistor, protože Darlingtonový pár bývá implementován v rámci jednoho pouzdra spolu s dalšími rezistory (viz Obr. 2.19), které stabilizují a urychlují spínání, a TVS diodou chránící tranzistor před poškozením při spínání zátěží s magnetickým polem (typicky induktivní zátěž).



Obr. 2.19: Interní zapojení Darlingtonového páru

Navržený sériový tranzistorový stabilizátor napětí je zobrazen na Obr. 2.20. Jeho funkčnost byla opět ověřena simulací v prostředí Micro-Cap. V prvním grafu na Obr. 2.21 je vidět, že výkonová ztráta rezistoru R1 s dostatečnou rezervou nepřesáhne maximální hodnotu 0,75 W a výkonová ztráta zenerovy diody ZD1 nepřekročí limitní hodnotu 1 W. Při návrhu bylo snahou, aby byl proud tekoucí ZD1 při 24 V dostatečně velký (cca 2 mA) a zároveň na ní nebyl velký ztrátový výkon při vyšším



Obr. 2.20: Sériový tranzistorový stabilizátor napětí

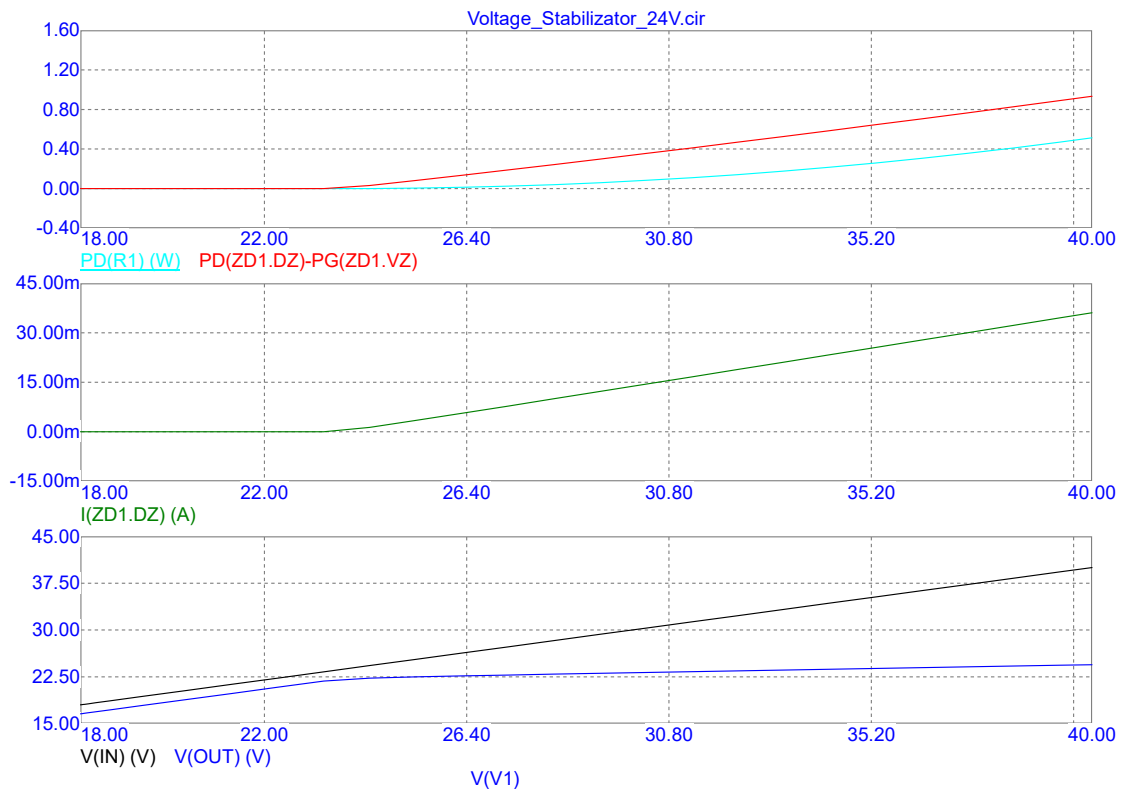
napětí. Průběh tohoto proudu lze pozorovat v druhém grafu. Třetí graf zobrazuje vztah mezi vstupním napětím (černý průběh) a výstupním napětím (modrý průběh). Lze vidět, že při vstupním napětí 24 V by bylo výstupní napětí přibližně 22,5 V. Tento pokles je dán napěťovým úbytkem na přechodu báze-emitor Darlingtonového zapojení tranzistoru Q1.

V předchozím odstavci byla popsána simulace navrženého sériového tranzistorového stabilizátoru napětí. Jeho simulační obvodové zapojení lze vidět na Obr. 2.22.

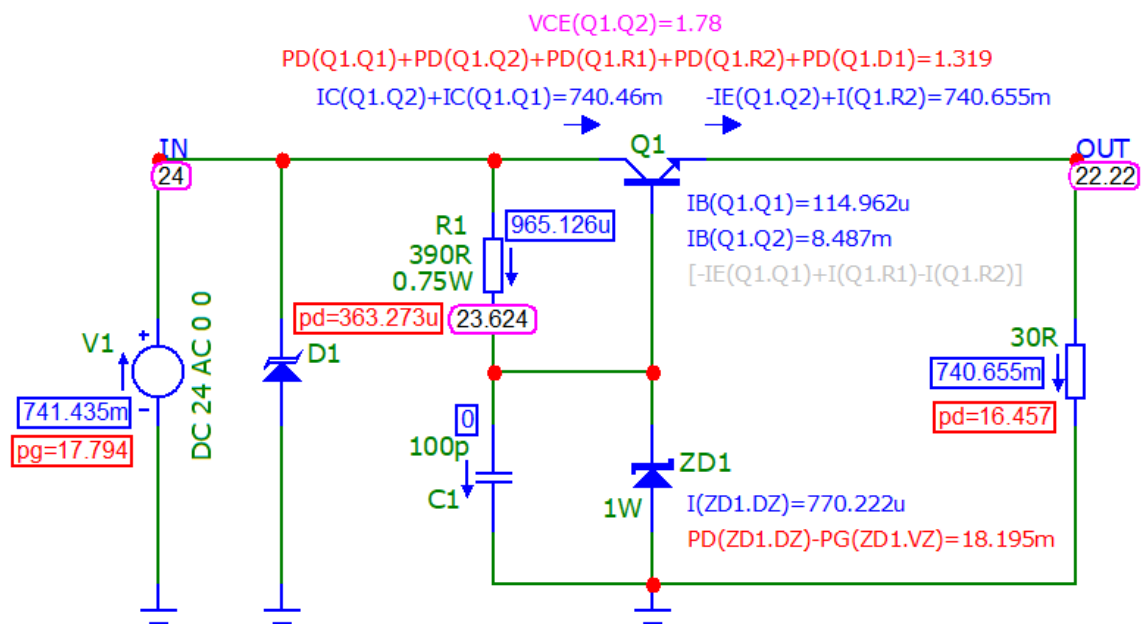
### Spínaný DC-DC měnič

Pro konverzi napětí ze stabilizovaných přibližně 22,5 VDC na 9 VDC byl použit spínaný DC-DC měnič R-78B9.0-2.0 od výrobce RECOM (viz Obr. 2.23). Měnič je navržen tak, aby byl snadno nahraditelný typickými lineárními regulátory typu 78xx. Umožňuje poměrně veliký rozsah vstupního napětí (11 až 32 VDC) a dosahuje poměrně slušné efektivity (čím nižší vstupní napětí, tím lepší). Odebíraný proud může nabývat až 2 A.

V referenčním zapojení DC-DC měniče se na vstupu měniče nachází EMC filtr (viz Obr. 2.23), který je navržen tak, aby se shodoval s elektromagnetickou kompatibilitou podle třídy B standardu EN55022.

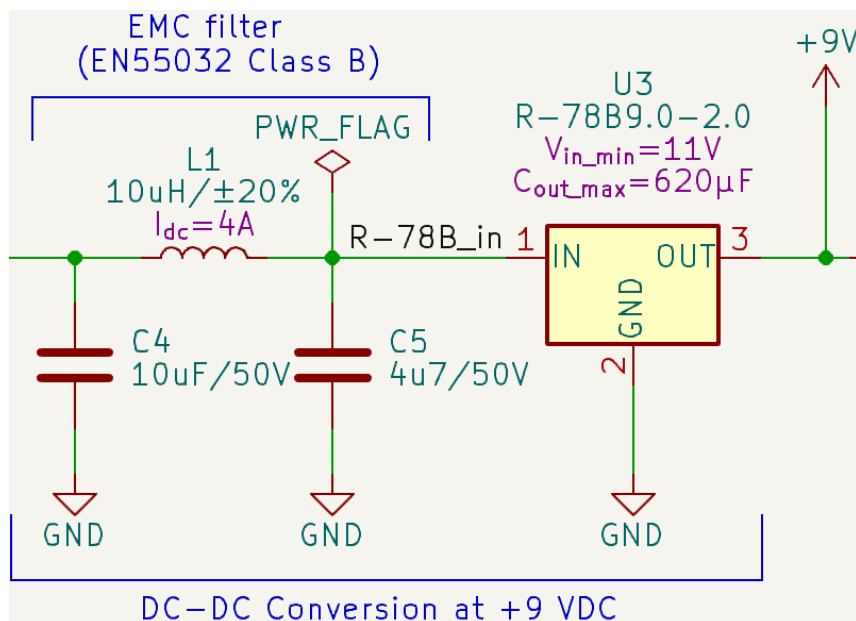


Obr. 2.21: Simulace sériového tranzistorového stabilizátoru napětí



Obr. 2.22: Simulační obvod sériového tranzistorového stabilizátoru napětí

Získaných 9 VDC bylo následně stabilizováno na 5 VDC pomocí lineárního regulátoru AMS1117-5.0. Toto napětí bylo využíváno pro napájení 8-kanálového ADC



Obr. 2.23: DC-DC měnič (U3) na +9 V s EMC filtrem (L1, C4, C5) na vstupu

a paměť EEPROM. Následně byla provedena další stabilizace na 3,3 VDC.

## 2.2 Testovací DPS

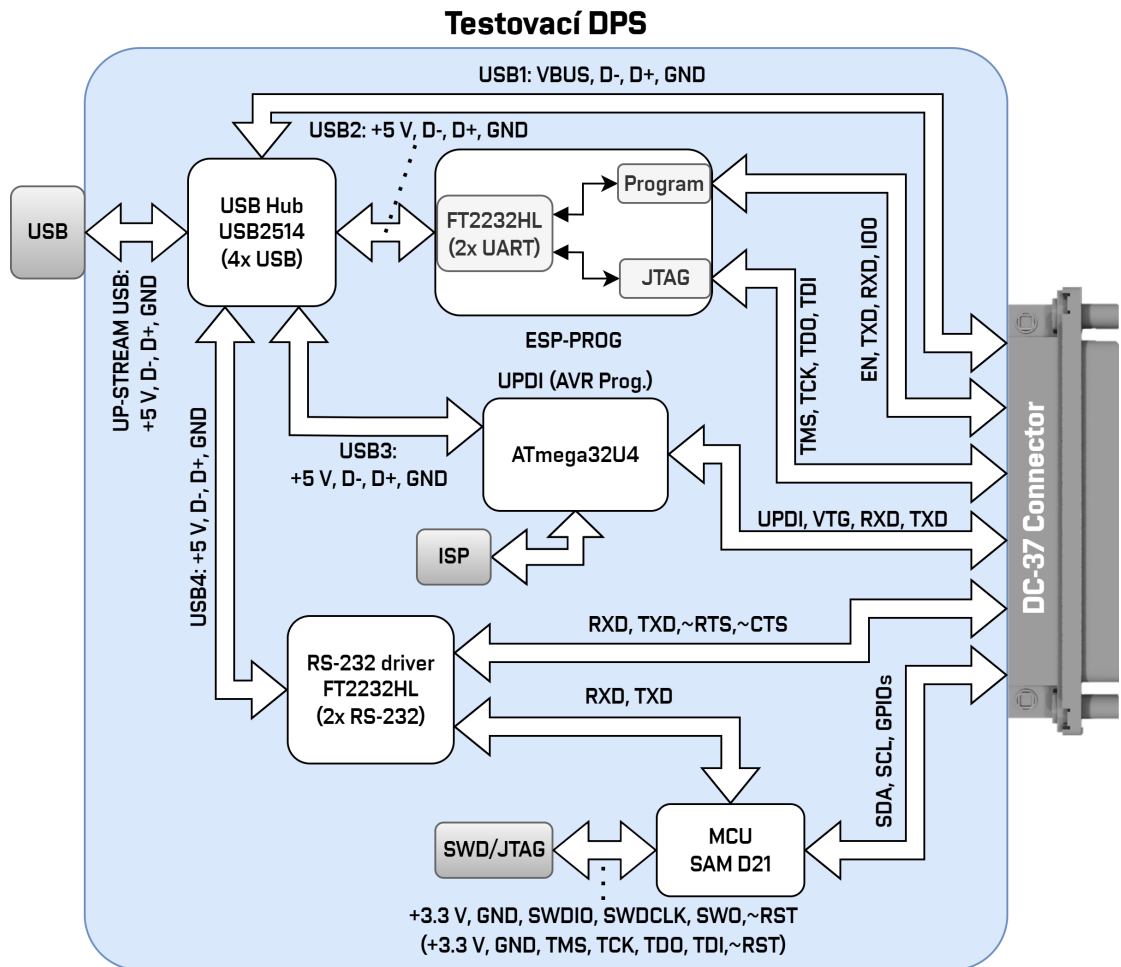
Testovací DPS si lze představit jako zprostředkovatele rozhraní pro komunikaci, programování, ladění a vykonavatele testovacích rutin. Její jádro je tvořeno mikroprocesorem ATSAM21G18A-AU od firmy Microchip, na kterém bude běžet vytvořený firmware komunikující s počítačovou aplikací. Hlavním účelem mikroprocesoru osazeného na testovací DPS je zpracovávat povely z desktopové aplikace a zpětně poskytovat informace o naměřených hodnotách v závislosti na aktuálně zvolené konfiguraci testované DPS.

Testovací deska zprostředkovává komunikačního rozhraní MCU mezi přizpůsobovací DPS (konektor D-Sub DC-37) a počítačem (USB) v rámci programovacích a ladících rozhraní (ESP-PROG, UPDI) pro možné varianty mikroprocesorů osazených na testované DPS. V současné době je předpokládáno, že na testované DPS bude osazen 8-bitový AVR mikroprocesor a/nebo modul ESP32. Na Obr. 2.24 lze vidět blokový diagram testovací desky.

### 2.2.1 Funkční bloky elektrické části návrhu

V následujících podsekcích jsou podrobněji popsány funkční bloky, které byly použity při návrhu elektrického schématu testovací DPS. Těmito funkčními bloky jsou





Obr. 2.24: Blokový diagram testovací DPS

rozbočovač pro vstupní USB rozhraní, převodník z USB na 2 sériové kanály, programátor pro moduly ESP32, programátor pro AVR mikroprocesory s vlastním programovacím rozhraním *In-circuit Serial Programming* (ISP), mikroprocesor SAM D21 včetně vlastního programovacího rozhraní a napájecí obvody.

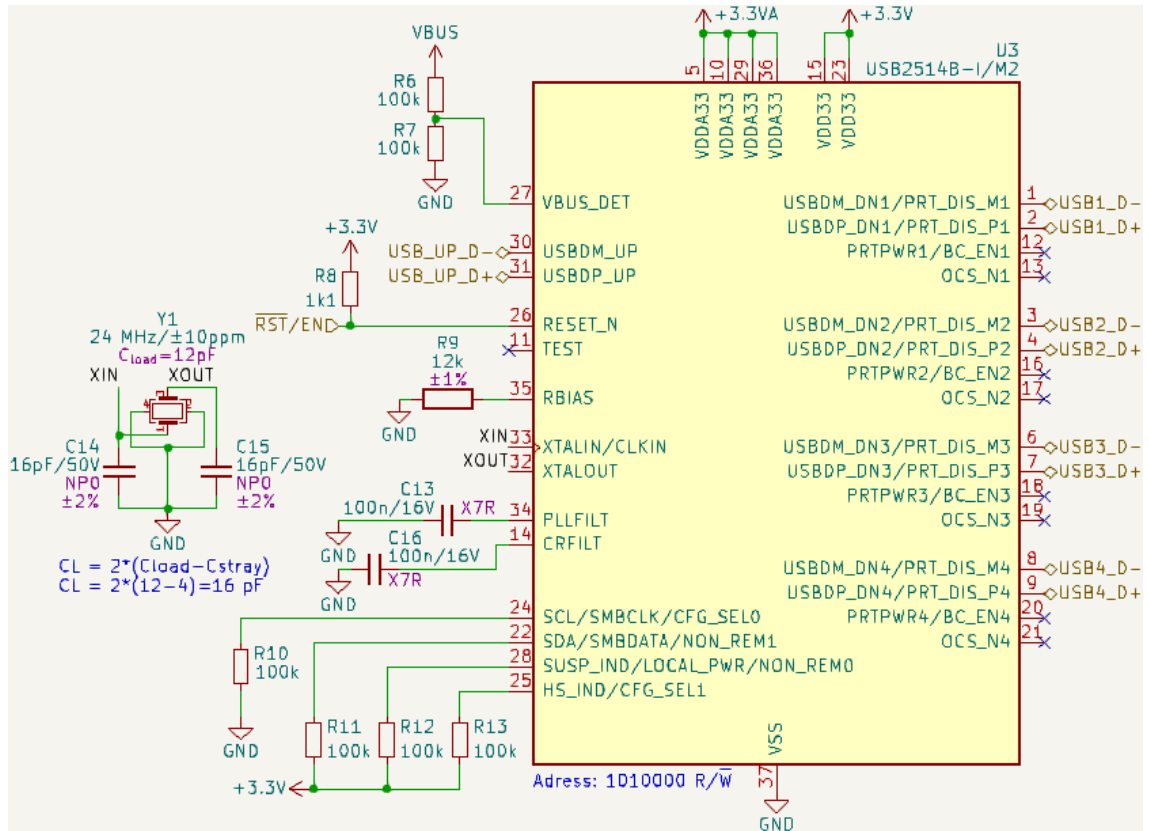
### Rozbočovač pro vstupní USB rozhraní

Jako rozhraní mezi testovací DPS a počítačem slouží USB, které bylo pomocí USB rozbočovače (*hub*) rozšířeno na čtyři nezávislé kanály USB. Pro tento účel byl vybrán čip USB2514B od firmy Microchip.

USB hub má vždy přesně jeden *upstream* port a více *downstream* portů. Port označovaný jako upstream je využíván pro připojení rozbočovače k hostiteli (*host*), a to přímo nebo přes další rozbočovače. Na výstupní porty hubu lze připojit další rozbočovače nebo koncová zařízení.

Při běžném přenosu se rozbočovače jeví transparentní – data přijatá na jejich

upstream portu jsou vyslána ke všem zařízením připojeným k jejich downstream portům. Opačně data přijatá na downstream portu jsou obecně předávána pouze směrem do upstream portu (výjimkou je signalizace obnovení). Ve standardu USB 3.0 bylo zavedeno směřování *point-to-point*, tedy hostiteli je umožněno směřovat odešlý paket na jeden cílový port, a tím je zajištěna nižší spotřeba energie a menší přetížení sítě [25].



Obr. 2.25: Schematické zapojení 4-kanálového USB hubu

Na Obr. 2.25 lze vidět použitý návrh zapojení 4-kanálového USB hubu. Na pravé straně symbolu pro čip USB3514B-I (U3) se nachází všechny 4 downstream kanály, jejichž konkrétní využití je názorně ukázáno na Obr. 2.24. Konfigurační (*strapping*) piny s funkcí BC\_EN[4:1] slouží k povolení bateriového nabíjení na jednotlivých downstream portech. Piny s funkcí OCS\_N[4:1] snímají příchod příznaku (*flag*) z externího čipu pro správu napájení jednotlivých downstream portů signalizující nadměrný proudový odběr. V rámci návrhu se nepočítá s budoucí potřebou nabíjení externích baterií ani s externími obvody pro správu napájení USB portů, a proto nejsou zmíněné piny využity.

Hub umožňuje pomocí pinů CFG\_SEL[1:0] nastavit, zda budou jednotlivé kanály napájeny z vlastních zdrojů nebo má být k napájení využit upstream port. Ten by měl být podle standardu schopný dodat až 0,5 A. K účelu napájení byl zvolen

upstream port a důvod této volby je podrobněji popsán na konci této podkapitoly. Ke kontrole přivedeného napájecího napětí slouží pin 27, který snímá hodnotu napětí přes odporový dělič 2:1.

Dostatečně přesný taktovací signál je zprostředkován externím krystalovým oscilátorem (Y1). Aby byla zajištěna jeho přesnost, je nutné na jeho vstup i výstup (XIN, XOUT) připojit kondenzátory, jejichž hodnotu lze vypočítat dle následujícího vztahu:

$$C_L = 2 \cdot (C_{load} - C_{stray}), \quad [\text{pF}] \quad (2.5)$$

kde  $C_{load}$  je “zatěžovací kapacita krystalu“, kterou lze najít v jeho datovém listu a  $C_{stray}$  je odhadovaná hodnota „rozptylové kapacity“ měděných cestiček vedoucích k čipu – obecně bývá odhadována na 2 až 6 pF.

Tedy:

$$C_L = 2 \cdot (12 - 4) \cdot 10^{-12} = 16 \text{ pF}.$$

Konfigurační piny NON\_REM[1:0] slouží k nastavení, zda budou downstream porty za provozu odjímatelné. Připnutí obou těchto pinů k napájecímu napětí přes pull-up rezistory zajistí, že odjímatelný bude pouze poslední kanál.

## Programátor modulů ESP32 (ESP-PROG)

Za účelem programování modulů ESP32 (na testované DPS) je testovací DPS vybavena obvodem, který byl převzat z volně dostupné dokumentace<sup>2</sup> k modulu ESP-PROG firmy Espressif. Uvnitř tohoto modulu je druhý kanál z USB rozbočovače rozdělen na dva nezávislé UART kanály (pomocí čipu FT2232HL – viz podsekcce Převod z USB na UART), přičemž první kanál je využit pro programování a druhý pro ladění (*debugging*) modulů ESP32.

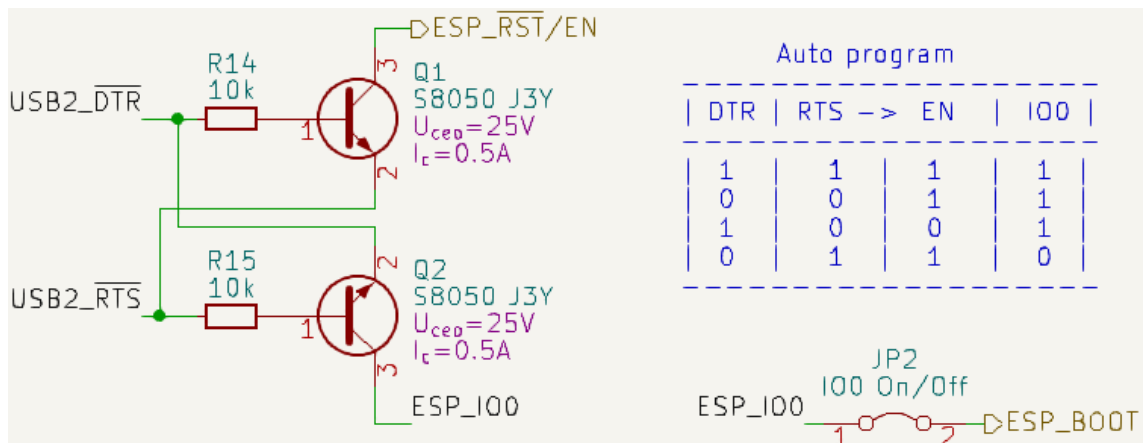
Obvod je dále vybaven tlačítky pro spouštění (*boot*) a resetování (*reset*) programovaného modulu. Funkci těchto tlačítek a zároveň automatický programovací režim (využívající signály  $\overline{DTR}$  a  $\overline{RTS}$ ) zastupují tranzistory Q1 a Q2. Aby bylo možné odpojit signál boot od automatického spouštěcího režimu, byla přidána pájecí propojka (*jumper*) JP2. Schematické zapojení lze vidět na Obr. 2.26.

Programované mikročipy často bývají navrženy na napájecí napětí 3,3 V nebo 5 V. Z tohoto důvodu jsou součástí obvodu nepájivé jumpery JP1 a JP3, které umožňují obsluze volit napájecí napětí pro programování a ladění podle aktuální potřeby.

Kompletní elektrické schéma programátoru modulů ESP32 bylo vzhledem ke své velikosti umístěno do příloh (viz Příloha B.2).

---

<sup>2</sup>Dostupné z: [https://www.espressif.com/sites/.../ESP-Prog\\_V2.1\\_reference\\_design\\_r1.0.zip](https://www.espressif.com/sites/.../ESP-Prog_V2.1_reference_design_r1.0.zip)



Obr. 2.26: Schematické zapojení tranzistorů (Q1, Q2) pro automatický programovací režim

### Programování AVR mikroprocesorů (UPDI)

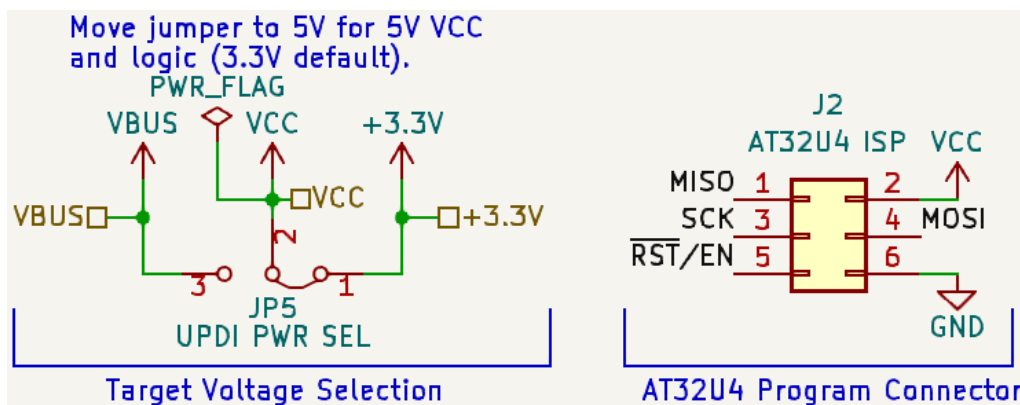
Třetí kanál z USB rozbočovače je přiveden přes rezistory 22R přímo na piny 3 a 4 mikroprocesoru ATmega32U4, který je využit jako tzv. mEDBG (*mini Embedded Debugger*). Jak již z názvu vyplývá, jedná se o ladicí nástroj, který je vestavěný přímo na DPS a není tedy nutné pořizovat a připojovat externí programátor. Originální mEDBG od firmy Microchip je schopen programovat a ladit mikročipy přes rozhraní ISP/dWire, UPDI, SWD či *Tiny Programming Interface* (TPI).

V rámci testovací DPS je využito rozhraní UPDI za účelem programování a ladění mikročipů megaAVR (či tinyAVR) na testované DPS. Pro signál UPDI je na ATmega32U4 vyhrazen pin 1 (PE6). Dále je k dispozici také UART rozhraní zajišťující komunikaci při ladění programovaného mikročipu. Signály TXD a RXD jsou navíc vedeny přes *Complementary Metal–Oxide–Semiconductor* (CMOS) buďdiče SN74LVC1G125 sloužící také jako vyrovnávací paměť.

ATmega32U4 byl dále vybaven třemi LEDkami – 2 (připojené na piny PD2 a PD3) signalizující probíhající komunikaci na TXD a RXD signálech a třetí (na pinu PC6) zobrazující aktuální stav mikročipu.

Aby bylo možné volit napájecí napětí cílového zařízení mezi 3,3 V a 5 V, byl navíc přidán selektivní nepájivý jumper (JP5). Pro zprovoznění a zajištění správné funkce mEDBG je nutné nahrát do ATmega32U4 oficiální programové vybavení (*firmware*) pro mEDBG od firmy Microchip. K tomu je vyhrazen I(C)SP konektor (J2), který lze vidět na Obr 2.27.

Z důvodu rozsahu elektrického schématu byl návrh programátoru pro AVR mikroprocesory umístěn do příloh (viz Příloha B.3).



Obr. 2.27: Schematické zapojení jumperu pro selekci napájecího napětí UPDI a ISP konektoru

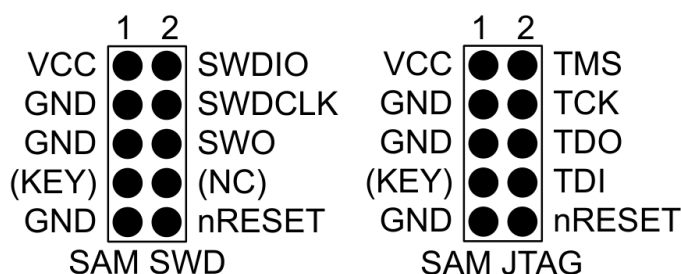
### Převod z USB na UART (RS-232)

Poslední kanál z USB rozbočovače je s využitím čipu FT2232HL dále převeden na 2 nezávislé kanály UART (RS-232), přičemž jeden slouží jako průchozí na přizpůsobovací DPS, kde je dále převeden na RS-485 pro komunikaci s testovanou DPS, a druhý je vyhrazen pro komunikaci s mikroprocesorem SAM D21 na testovací DPS.

Převodník FT2232HL také využívá externí krystalový oscilátor jako zdroj přesného taktovacího signálu. Z důvodu signalizace probíhající komunikace na průběžném RS-232 do přizpůsobovací desky byly přidány 2 LED diody (pro signály RX a TX). Zapojení elektrického návrhu je umístěno do Přílohy B.4.

### Programovací rozhraní SWD/JTAG

Rozhraní SWD je na testovací DPS vyhrazeno pro programátor mikroprocesoru SAM D21. Ačkoli pro programování po SWD stačí pouze 2-3 (plus napájení a země) vodiče, byl pro připojení vybrán 10-ti pinový konektor kvůli kompatibilitě s rozhraním JTAG. Rozložení pinů pro zmíněná 2 rozhraní lze vidět na Obr. 2.28. Jako programátor byl zvolen nástroj Atmel-ICE pro svou univerzálnost a spolehlivost.



Obr. 2.28: Doporučené rozložení programovacího konektoru pro ARM SWD/JTAG (Převzato z: [https://ww1.microchip.com/.../Atmel-ICE\\_UserGuide.pdf](https://ww1.microchip.com/.../Atmel-ICE_UserGuide.pdf))

## Napájecí obvod

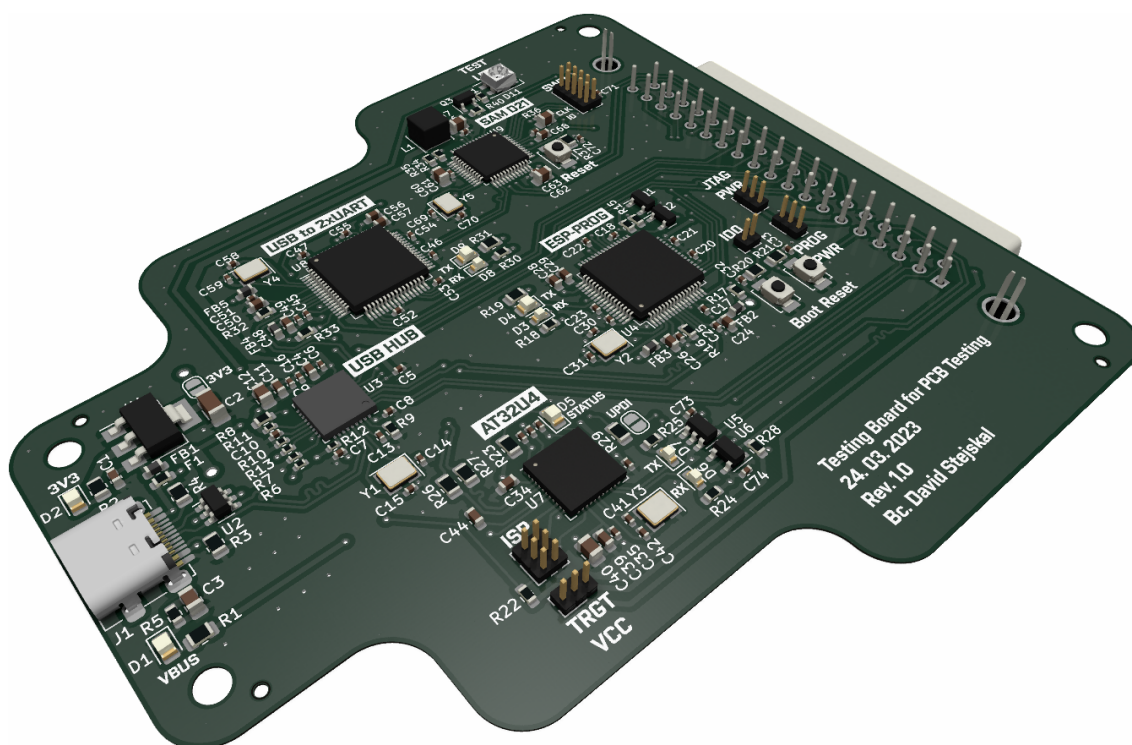
Napájecí napětí testovací DPS je přivedeno společně se sériovou komunikací po USB. Standard USB-C umožňuje proudový odběr až 0,5 A ze zdrojového zařízení, který s dostatečně velkou rezervou vystačí pro napájení testovací i přizpůsobovací desky. Elektrické schéma jištění a filtrace napájecího obvodu lze vidět v Příloze B.1.

### 2.2.2 Popis funkce a vlastností vybraných komponent

V Tab. 2.3 je uveden výběr podstatných komponent použitých v návrhu testovací DPS. Ke každé komponentě je v druhém sloupci tabulky uveden také stručný souhrn parametrů.

### 2.2.3 Finální výstup návrhu testovací desky

Obdobně jako u přizpůsobovací DPS, v této kapitole jsou prezentovány finální výstupy navržené testovací desky. Opět bylo využito nástroje 3D Viewer softwaru Kicad 7.0 k zobrazení a exportu 3D pohledů. Pohled shora lze vidět na Obr. 2.29 a zespodu na Obr. 2.30.



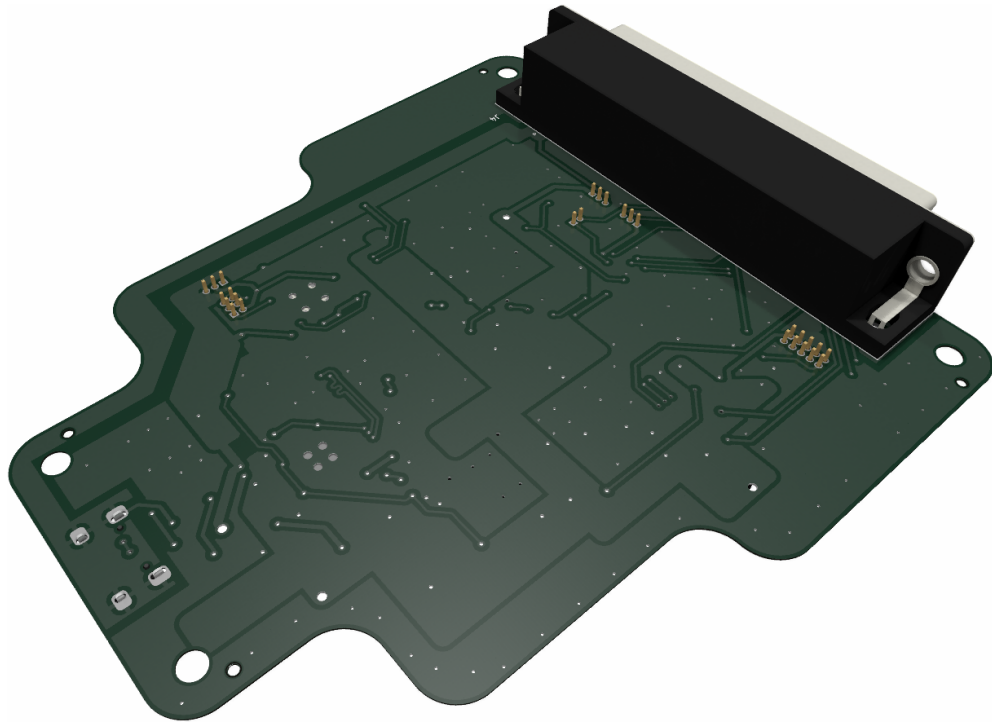
Obr. 2.29: 3D model testovací DPS – pohled shora po úhlem

Vytvořený výkres měděných ploch vrchní vrstvy DPS včetně okrajů DPS a pisků je ukázán na Obr. 2.31.

Tab. 2.3: Podstatné vybrané komponenty na testovací DPS

Komponent	Stručný popis
ATSAMD21G18A-AU	32-bitový mikroprocesor rodiny SAMD21 – založen na ARM® Cortex®-M0+, pracovní frekvence až 48 MHz, velikost paměti pro program (FLASH) až 256 KB, RAM paměť 32 KB, 38 I/O, 14x12-bitový ADC, 1×10-bitový <i>Digital to Analog Converter</i> (DAC), 20× <i>Pulse Width Modulation</i> (PWM), 6 flexibilních sériových komunikačních modulů (SERCOM), <i>Direct Memory Access</i> (DMA) a <i>event</i> systém, napájecí napětí 1,62 až 3,6 V
ESP-PROG	vývojový, programovací i ladící modul pro ESP32 – umožňuje volbu mezi napájením 3,3 V a 5 V, zajišťuje převod z USB na rozhraní JTAG (ladění) i sériovou komunikaci pro programování pomocí čipu FT2232HL, možnost výběru režimu spouštění
UPDI	programovací i ladící rozhraní pro 8-bitové AVR mikroprocesory – proprietární jednodrátové rozhraní společnosti Microchip, externí programování a ladění zařízení přímo na čipu, poskytuje obousměrnou poloduplexní asynchronní komunikaci, nástupce dvou vodičového fyzického rozhraní <i>Program and Debug Interface</i> (PDI) (nacházející se na všech zařízeních AVR XMEGA)
FT2232HL	2-kanálový RS-232 řadič – převod z USB na 2 nezávislé kanály rozhraní UART (RS-232), podporuje high/full-speed USB, napájecí napětí 3,3 V
USB2514B	4-kanálový USB 2.0 Hub (rozbočovač) řadič – konfigurovatelný, <i>Multi Transaction Translator</i> (MTT) <i>hub</i> , podporuje low/full/high-speed <i>downstream</i> zařízení, plně integrované USB zakončení i <i>pull-up/pull-down</i> rezistory, napájecí napětí 3,3 V, integrovaný/externí 24 MHz oscilátor

Taktéž u testovací DPS byla řízena zakončovací (50  $\Omega$ ) i vzájemná (90  $\Omega$ ) impedance diferenciálních párů a stejně tak byla kontrolována jejich délka společně s komunikačními vodiči (TX, RX) sériových linek. To vše v mnohem větším měřítku než u přízpůsobovací DPS kvůli jedné z hlavních podstat testovací DPS, kterou je zajištění komunikace jednotlivých programovacích i ladících rozhraní.



Obr. 2.30: 3D model testovací DPS – pohled zespodu po úhlem

Výrobní výkres s umístěním komponent včetně pájecí masky byl taktéž umístěn do příloh (viz Příloha C.2). Zde je patrné, že při rozmístování komponent byl kladen důraz na elementární i pokročilá návrhová pravidla.

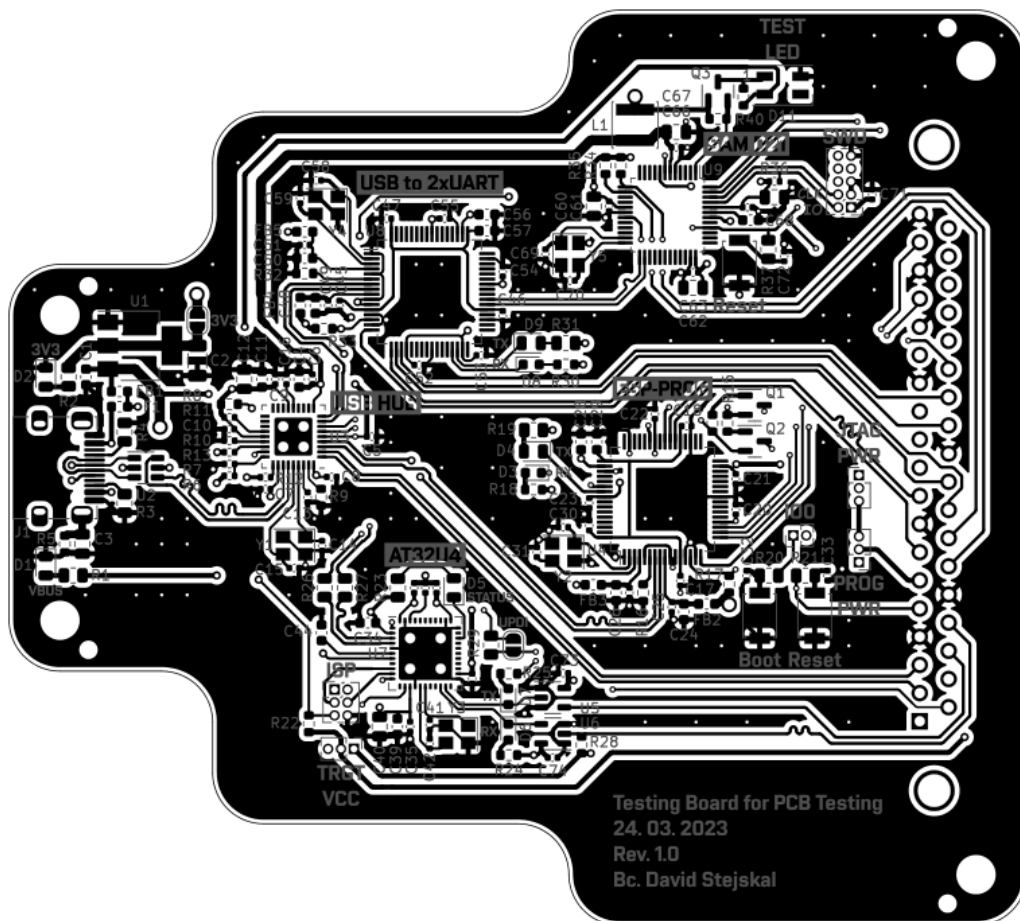
Blokovací kondenzátory jsou vždy umístěny v dostatečné blízkosti příslušejícímu čipu, resp. konkrétnímu napájecímu pinu (viz Obr. 2.32). Taktéž je využíváno kombinací různých pouzder (0402, 0603, 0805) a hodnot kapacit (100n, 1u, 2u2, 4u7, 10u) těchto kondenzátorů, aby bylo dosaženo co možná nejefektivnějšího eliminování poklesu napájecího napětí při nárazovém proudovém odběru během překlápění logických obvodů daného čipu. Zároveň je tak zajištěno výrazné potlačení vysokofrekvenčního rušení, protože kondenzátory s nižšími hodnotami (např. 100n) představují zkrat pro vysoké frekvence [22].

U konektoru USB-C si lze všimnout použité ochrany proti elektrostatickému výboji a přepětí (USBLC6-2SC6), dále byla přidána také vratná pojistka kvůli omezení proudu, resp. odpojení v případě zkratu (viz Obr. 2.33).

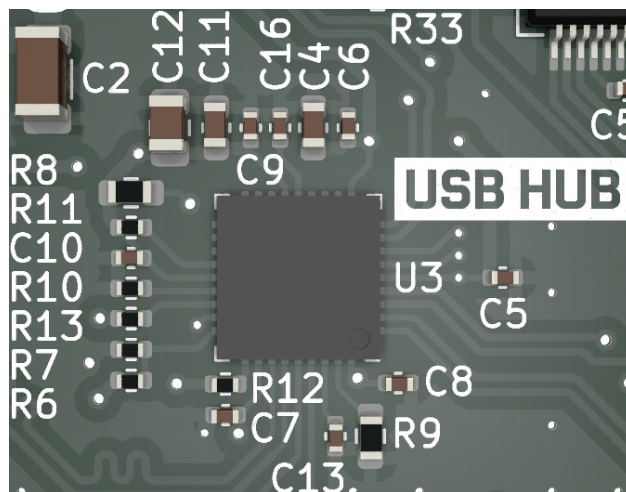
## 2.3 Rozhraní mezi deskami

V Tab. 2.4 je uvedeno rozdělení 37 vodičů konektoru D-Sub DC-37, který tvoří rozhraní mezi testovací a přizpůsobovací deskou. Posledních 6 vodičů je využito pro Reset signál a 5-krát PWM výstup z MCU pro případ potřeby v budoucnu.



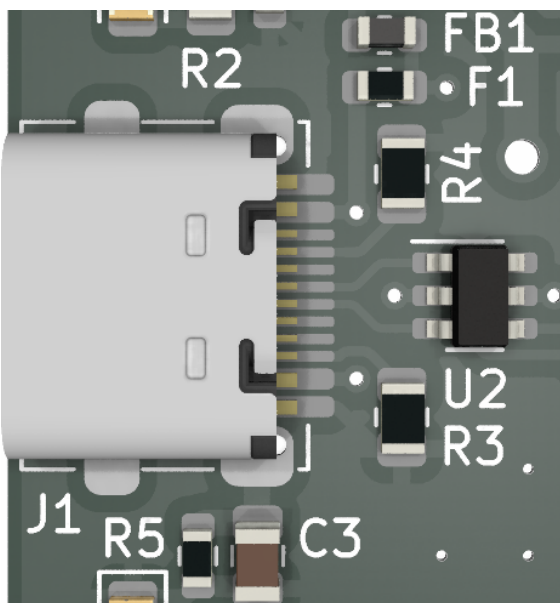


Obr. 2.31: Vrchní vrstva testovací DPS – měděná vrstva s potiskem



Obr. 2.32: Ukázka rozmístění blokovacích kondenzátorů USB hubu

V rámci rozhraní pro programování a ladění ESP modulů přes JTAG byly zachovány signály GND, které mohou být nápomocné při eliminaci přeslechu mezi vodiči v případě připojení desek pomocí 37-žilového kabelu. Nicméně vzhledem k malé



Obr. 2.33: Jištění (U2, F1) a filtrace (FB1) napájecího obvodu na DPS

Tab. 2.4: D-Sub DC-37 konektor – rozhraní mezi testovací a přizpůsobovací DPS

USB				ESP-PROG: Program					
1	2	3	4	5	6	7	8	9	10
VBUS	DN	DP	GND	EN	VDDP	TXD	GND	RXD	IO0

ESP-PROG: JTAG									
11	12	13	14	15	16	17	18	19	
VDDJ	TMS	GND	TCK	GND	TDO	GND	TDI	GND	

UPDI				I2C		
20	21	22	23	24	25	26
UPDI	VTG	TXD	RXD	GND	SDA	SCL

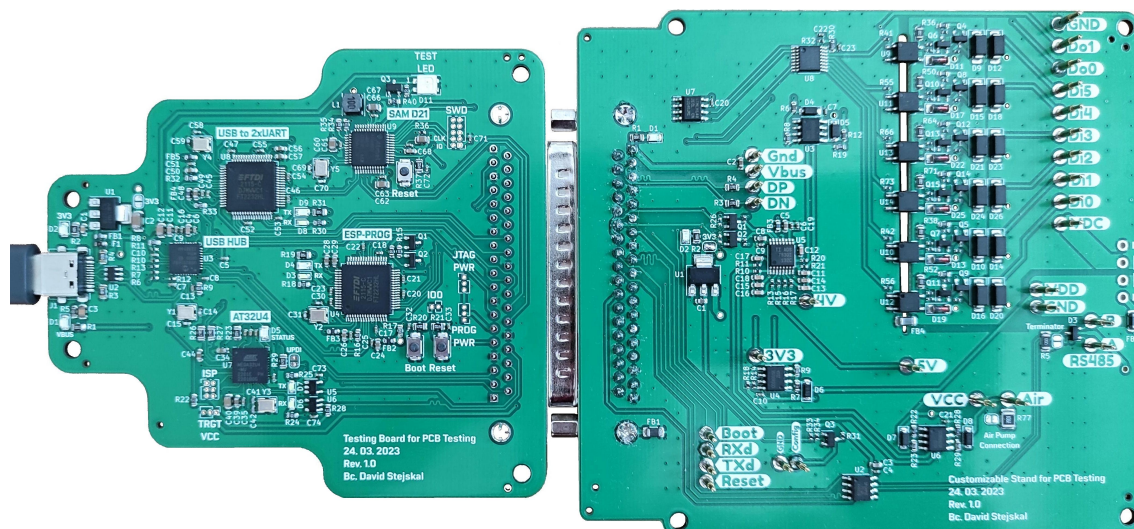
  

RS-232					SAM D21					
27	28	29	30	31	32	33	34	35	36	37
RXD	TXD	GND	RTS	CTS	RST	PB02	PB03	PB08	PB09	PB10

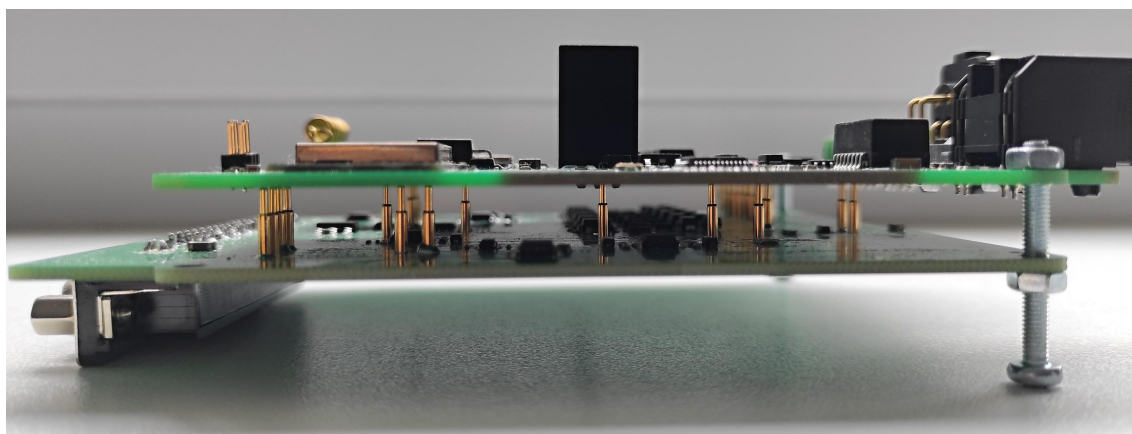
pravděpodobnosti tohoto připojení by do budoucna stálo za zvážení zredukovat příp. nahradit tyto signály za jiné.

## 2.4 Kompletace testovacího zařízení

Tato podkapitola byla vytvořena za účelem prezentovat výsledné hardwarové řešení první verze testovacího zařízení. Na Obr. 2.34 lze vidět propojení testovací a přizpůsobovací DPS pouze pomocí D-Sub DC-37 konektoru a na Obr. 2.35 je demonstrováno umístění testované DPS na tzv. testovací podstavec (přizpůsobovací DPS).



Obr. 2.34: Testovací zařízení – propojení testovací a přizpůsobovací DPS



Obr. 2.35: Testovací zařízení – propojení přizpůsobovací a testované DPS

## 3 Návrh a implementace SW

Tato kapitola podrobně pojednává o tvorbě programového vybavení pro testovací stanici využívající mikročip SAM D21, který zajišťuje komunikaci s desktopovou aplikací za účelem vykonávání automatických testů, vizualizaci výsledné zprávy ad.

### 3.1 Microchip Studio

Při návrhu a implementaci programového vybavení (*firmwaru*) testovací stanice bylo využito editoru kódu Microchip Studio verze 7.0.2594 (dříve Atmel Studio). Tento editor byl vybrán z důvodu nejlepší podpory čipů a vývojových desek od společnosti Microchip a zároveň vzhledem k doposud získaným zkušenostem během navazujícího studia. S využitím online nástroje Atmel START, který umožňuje grafickou konfiguraci projektu včetně výběru i konfigurace softwarových komponent a ovladačů, je tvorba softwaru pro produkty Microchip mnohem pohodlnější a rychlejší.

### 3.2 Tvorba ovladačů pro použité periferie

Jak již bylo uvedeno v podkapitole 2.1.1, na přizpůsobovací DPS se nachází následující komponenty komunikující po sběrnici I2C: oktalový GPIO expandér, 8-kanálový A/D převodník a EEPROM. Pro každý z nich bylo nutné vytvořit ovladač (*driver*) v jazyce C obsahující potřebné funkce.

#### 3.2.1 Ovladač pro oktalový GPIO expandér

GPIO expandér je užitečný komponent pro vzdálené ovládání/čtení digitálních výstupů/vstupů po pouhých dvou signálových vodičích (v případě sběrnice I2C). Byla použita oktalová verze, přičemž ze všech osmi kanálů je IO0 nevyužit, IO1 použit jako vstup pro konfigurační signál z MCU testovací DPS (vysoká úroveň na tomto vstupu „přízemní“ POGO pin Config, který testované DPS signalizuje, že je v režimu konfigurace) a ostatní IO2-IO7 jsou využity jako digitální výstupy ovládající výkonné stupně (viz Obr. 2.4) – tyto obvody jsou navrženy v pozitivní logice, takže nízká úroveň znamená GND1 a vysoká úroveň VDC na výstupu.

GPIO expandér disponuje čtyřmi registry: registr 0 – vstupní port (adresa 0x00), registr 1 – výstupní port (adresa 0x01), registr 2 – inverze polarity (adresa 0x02) a registr 3 – konfigurační registr (adresa 0x03). V rámci každého začátku příkazu expandéru tak musí po jeho adrese (včetně bitu definující operaci  $\overline{W}$  (*Write*)) vždy následovat byte obsahující adresu registru, se kterým je v úmyslu pracovat. **Vstupní port** slouží pro čtení logických hodnot na vstupních pinech a je jako jediný pouze

pro čtení (*read only*) – zápis do něj nemá žádný vliv. **Výstupní port** reflektuje logické úrovně pinů, které jsou konfiguračním registrem definovány jako výstupy. Čtením z tohoto portu nejsou získány aktuální hodnoty pinů, ale hodnoty vnitřních klopných obvodů (*flip flop*) řídicí výběr výstupu. **Registr inverze polarity** umožňuje využití inverzní logiky u pinů, které jsou v konfiguračním registru definovány jako vstupy. Vysoká logická úroveň pro konkrétní pin v tomto registru inverzi povoluje a nízká inverzi ruší. **Konfigurační registr** určuje, který pin představuje digitální vstup (log. 1) či digitální výstup (log. 0).

V rámci ovladače tak byly implementovány funkce uvedené ve Výpisu 3.1. Podle názvu funkce lze také rozlišit (podle podtržítka na začátku jména), zda se jedná o funkci privátní (interní) nebo veřejnou (k dispozici při používání ovladače). Interní funkce jsou základními stavebními kameny funkcí veřejných.

Výpis 3.1: Funkce ovladače (pca9557) pro oktalogý GPIO expandér

```

93 // private functions:
94 uint8_t _pca9557_get_reg(pca9557_reg_addr_t reg_addr);
95 uint8_t _pca9557_get_pin(pca9557_pin_t pin,
    pca9557_reg_addr_t reg);
96 ERROR_t _pca9557_set_reg(pca9557_reg_addr_t reg, uint8_t
    reg_setting);
97 ERROR_t _pca9557_set_pin(pca9557_pin_t pin,
    pca9557_reg_addr_t reg, pca9557_state_t new_pin_val);
98
99 // public functions:
100 void pca9557_init(void);
101 void pca9557_reset(void);
102 uint8_t pca9557_get_pin_mode(pca9557_pin_t pin);
103 uint8_t pca9557_get_pin_state(pca9557_pin_t pin);
104 uint8_t pca9557_get_pin_polarity(pca9557_pin_t pin);
105 ERROR_t pca9557_set_pin_mode(pca9557_pin_t pin,
    pca9557_mode_t set_mode);
106 ERROR_t pca9557_set_mode_all(pca9557_mode_t set_mode);
107 ERROR_t pca9557_set_pin_state(pca9557_pin_t pin,
    pca9557_mode_t new_state);
108 ERROR_t pca9557_set_state_all(pca9557_mode_t new_state);
109 ERROR_t pca9557_set_pin_polarity(pca9557_pin_t pin,
    pca9557_polarity_t new_polarity);
110 ERROR_t pca9557_set_polarity_all(pca9557_polarity_t
    new_polarity);

```

### 3.2.2 Ovladač pro 8-kanálový A/D převodník

A/D převodník je neodmyslitelnou součástí každého systému, po kterém je vyžadováno měření analogové veličiny – v tomto případě elektrického napětí. Obdobně jako u GPIO expandéru, i zde je využito výhody měření všech osmi kanálů po dvou vodičích sběrnice I2C. Pochopitelně existují i varianty s 16-ti kanály a mnohem větším rozlišením (použitý A/D převodník má rozlišení pouze 8 bitů), avšak pro tento případ je 8 kanálů i 8-bitů rozlišení dostačujících.

Při zahájení každého příkazu pro ADC následuje po jeho adrese (včetně bitu operace  $\overline{W}$ ) příkazový byte (*command byte*). Ten definuje, v jakém módu má být měřeno (*Single-ended* (SE)/diferenciálně), na kterém kanálu má být měřeno (v případě diferenciálního měření jsou vybrány 2 kanály) a použitý mód napájení včetně volby zapnutí/vypnutí interní reference pro měření.

Ovladač pro používání 8-kanálového A/D převodníku disponuje funkcemi uvedenými ve Výpisu 3.2. Opět je využito rozdělení funkcí na privátní/veřejné. Měřicí obvody byly v rámci přizpůsobovací DPS navrženy tak, aby nebylo zapotřebí měřit diferenciálně, a proto byly zatím implementovány funkce pouze pro měření v módu SE. Funkce `ads7830_measure_single_ended` změří jeden konkrétní kanál, zatímco funkce `ads7830_measure_all_channels_SE` změří postupně všechny kanály v módu SE.

Výpis 3.2: Funkce ovladače (`ads7830`) pro 8-kanálový A/D převodník

```
93 // private functions
94 void _set_ads7830_i2c_cmd_sd_mode(ads7830_sd_mode_t
    sd_mode);
95 void _set_ads7830_i2c_cmd_pd_mode(ads7830_pd_mode_t
    pd_mode);
96 bool _set_ads7830_i2c_cmd_byte(ads7830_sd_mode_t sd_mode,
    uint8_t channel, ads7830_pd_mode_t pd_mode);
97
98 // public functions
99 void ads7830_init(void);
100 void set_ads7830_i2c_address(uint8_t i2c_address);
101 uint8_t get_ads7830_i2c_address(void);
102 ERROR_t ads7830_measure_single_ended(uint8_t channel,
    ads7830_pd_mode_t pd_mode, uint8_t *adc_data);
103 ERROR_t ads7830_measure_all_channels_SE(ads7830_pd_mode_t
    pd_mode, uint8_t *adc_data);
```

### 3.2.3 Ovladač pro EEPROM

EEPROM jsou ve vestavných (*embedded*) systémech velmi často využívaným komponentem, jednoduše proto, že umožňují ukládat data, která mohou být uchována např. z důvodu většího komfortu uživatele (nemusí stále dokola vyplňovat ty samé informace) nebo k uložení informativních dat o produktu od výrobce.

Velikost celého adresovacího prostoru je 256 kb, k jehož adresaci je zapotřebí 15-ti adresových bitů. Kapacita jednoho bloku paměti (celkem je jich 8 – každému odpovídá konkrétní adresa čipu – piny  $A_2$ ,  $A_1$ ,  $A_0$ ) je 32768 bitů. Proto každý příkaz pro EEPROM začíná její adresou (včetně bitu operace  $\overline{W}$ ) a následují 2 B k určení adresy pro čtení/zápis (výjimkou je čtení z aktuální adresy).

V rámci ovladače pro EEPROM byly implementovány funkce pro zápis 1 B nebo celé strany (64 B) na konkrétní adrese, čtení 1 B z aktuální adresy, čtení 1 B z konkrétní adresy a sekvenční čtení konkrétního počtu bytů (parametr *length*) z konkrétní adresy (viz Výpis 3.3).

Výpis 3.3: Funkce ovladače (at24c) pro EEPROM

```
87 // private functions
88 void _set_at24c_params(AT24C_t *i2c_eeprom);
89
90 // public functions
91 void at24c_init(void);
92
93 ERROR_t at24c_byte_write(uint16_t byte_address, uint8_t
    *byte_buffer);
94 ERROR_t at24c_page_write(uint16_t address, uint8_t
    *page_buffer, uint8_t length);
95
96 ERROR_t at24c_current_addr_read(uint8_t *byte_buffer);
97 ERROR_t at24c_random_read(uint16_t address, uint8_t
    *byte_buffer);
98 ERROR_t at24c_sequential_read(uint16_t address, uint8_t
    *data_buffer, uint16_t length);
```

## 4 Obslužná aplikace

V této kapitole je stručně popsán postup při návrhu desktopové aplikace pro testovací stanici. K psaní kódu byl využit velmi známý editor Visual Studio Code (viz sekce 4.1.1) od společnosti Microsoft. Samotné jádro pro tvorbu aplikace tvoří *framework* Qt od společnosti Qt Group. Qt je ve světě velmi populární především díky své univerzálnosti, použitelnosti na většině platform, rychlosti a schopnosti efektivně vytvářet velmi kvalitní a skvěle vypadající uživatelská rozhraní či aplikace. Křivka učení se s tímto frameworkem je sice z počátku poměrně mírná oproti např. Kivy či Tkinter, avšak při vývoji náročnějších a pohlednějších aplikací se zdá být Qt vhodnější (více v sekci 4.1.2).

### 4.1 Softwarové nástroje

V rámci této podkapitoly budou zmíněny a popsány softwarové nástroje a prostředí, kterých bylo využito při návrhu a implementaci desktopové aplikace. Na volbu jednotlivých nástrojů nebyly ze strany vedoucího kladeny žádné požadavky, a proto bylo možné vybírat čistě dle osobních preferencí. Především se jedná o editor Visual Studio Code, framework Qt pro tvorbu aplikací, nástroj Poetry spravující závislosti pro Python, databázový systém PostgreSQL a nástroj InstallForge pro generování instalačního souboru aplikace.

#### 4.1.1 Visual Studio Code

Mezi programátory nejčastěji používané kódové editory se bezesporu řadí Visual Studio Code (VSCode). Jedná se o prostředí, které je zcela zdarma a dostupné pro uživatele operačních systémů Windows, macOS i Linux. Samotný VSCode se řadí mezi *light-weight* programy, avšak díky své obrovské podpoře formou různých rozšíření (*extensions*) si jej lze přizpůsobit podle svých potřeb a stává se tak velmi výkonným a uživatelsky přívětivým. Vývojáři tak stačí se důkladně naučit pracovat s jedním nástrojem a nemusí se neustále zdlouhavě učit v nových programovacích prostředích například při přechodu k jinému programovacímu jazyku nebo při změně výrobce mikročipů. Ke tvorbě desktopové aplikace pro testovací stanici byla použita verze 1.77.0.

#### Python

Python rozšíření od společnosti Microsoft dělá VSCode skvělým editorem pro programování v Pythonu a funguje na jakémkoli operačním systému. Poskytuje automatické dokončování a IntelliSense, dále *linting* (*Pylance*), ladění, navigování v kódu,



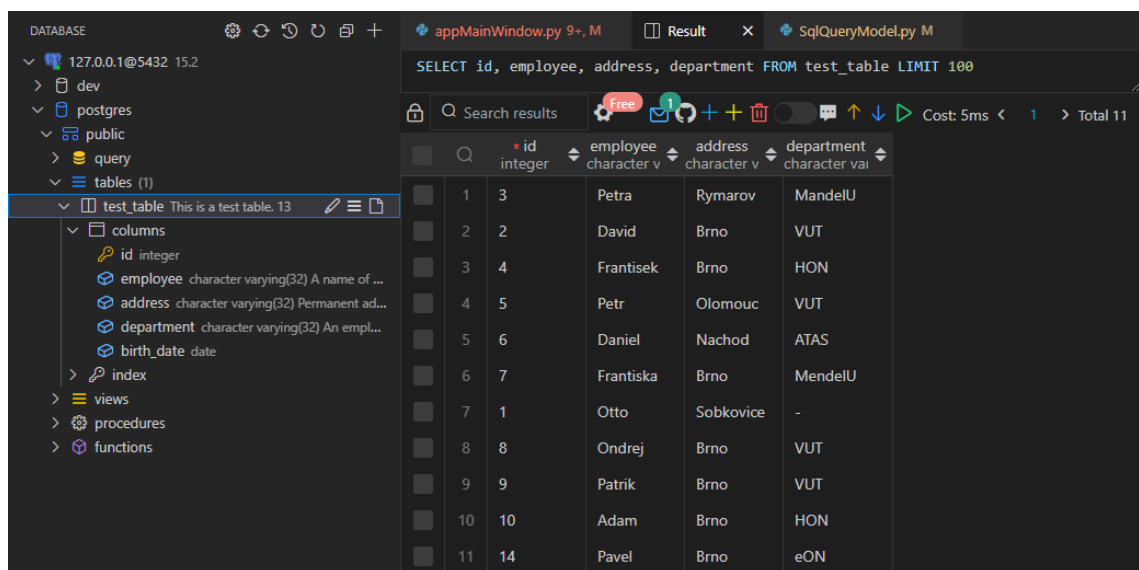
formátování kódu, refaktorování, testování (*unit tests*) a mnoho dalšího. Podporuje také snadné přepínání mezi prostředími Pythonu a efektivně rozpoznává přítomnost virtuálního prostředí (Poetry, Conda...) [9].

## Qt for Python

Qt for Python je název rozšíření pro VSCode, které zajišťuje pohodlnou práci při vývoji Qt aplikací v Pythonu. Toto rozšíření velmi úzce souvisí s oficiálním Qt for Python popsáném v podkapitole 4.1.2.

## PostgreSQL

V rámci desktopové aplikace bylo potřeba implementovat základní příkazy SQL databáze, známé jako *vytvořit, číst, aktualizovat, odstranit* – *Create, Read, Update, Delete* (CRUD). K otestování, zda operace s databází fungují správně, bylo použito rozšíření PostgreSQL (verze 6.3.5), které umožňuje nahlížení do tabulek v databázích, vytváření nových tabulek i záznamů a mnoho dalšího. Na Obr. 4.1 lze vidět rozhraní tohoto rozšíření při ladění funkčnosti příkazů.



Obr. 4.1: PostgreSQL rozšíření pro VSCode

### 4.1.2 Qt

Qt je multiplatformní software od společnosti Qt Group navržený k vytváření grafických uživatelských rozhraní (*Graphical User Interfaces* (GUI)) a aplikací, které mohou běžet na všech hlavních desktopových platformách, jimiž jsou například Windows, macOS či Linux a na většině mobilních nebo vestavěných platformách.

Qt je k dispozici kromě komerční licence také pod licencemi *open source*: GPL 2.0, GPL 3.0 a LGPL 3.0 [24].

Uživatelská rozhraní či aplikace vytvořené pomocí Qt mají nativní rozhraní, které je obecně nazýváno *widget*. Bez obtíží se dají vyvíjet také programy bez grafického rozhraní, jimiž jsou například nástroje příkazového řádku či konzole [24].

Qt podporuje spoustu běžných kompilátorů a navíc poskytuje Qt Quick obsahující *Qt Modeling Language* (QML), který umožňuje použití JavaScriptu pro poskytnutí potřebné logiky [24].

Qt poskytuje také funkce pro přístup k databázi SQL, parsování (syntaktickou analýzu) *JavaScript Object Notation* (JSON) či *Extensible Markup Language* (XML), správu vláken i podporu sítě [24].

*Qt for Python* (verze 6.5) je projekt, který poskytuje oficiální vazby (*bindings*) Pythonu pro Qt, protože Qt je vytvořené v C++. To jednoduše umožňuje Python vývojářům používat Python k psaní aplikací postavených na Qt. Aktuálně podporované verze jsou PySide6, PySide2, PyQt6 a PyQt5.

### 4.1.3 InstallForge

InstallForge je velmi nápomocný a volně dostupný nástroj pro vývojáře desktopových aplikací. Umožňuje totiž z navržené aplikace vytvářet instalační balíčky (formát ZIP) nebo dokonce „jedno-souborové“ instalátory (pro Windows typicky .exe soubory). Při vytváření instalátoru umožňuje nastavení spousty možností. Do stručného výběru lze zahrnout například přidání různých dialogových oken pro uživatele v průběhu instalace, nabídnutí uživateli volbu jazykových variant, možnost zvolit si, zda bude aplikace viditelná v nabídce Start (Windows) či sekci „Aplikace a funkce“, kde může být později odinstalována a mnoho dalšího. Za zmínku stojí i podpora Visual Update Express, která umožňuje provádět online aktualizace aplikace.

Aby bylo možné vytvořit instalační soubor, je potřeba nejdříve vygenerovat potřebná data z již navržené aplikace<sup>1</sup>. K tomu byl použit balíček `PyInstaller` pro Python. Po jeho nainstalování je nutné se přesunout do kořenového adresáře aplikace a zde z příkazové řádky spustit příkaz:

```
pyinstaller <hlavni-soubor-aplikace>.py
```

Po vykonání příkazu se ve stejném adresáři objeví složky `build` a `dist` (zkratka pro *distribution*). Ve složce `build` jsou shromážděna data s různými analýzami či hlášeními a na základě nich jsou následně vygenerovány potřebné soubory k samotné distribuci (složka `dist`). Právě tato data jsou potřebným výstupem pro vytvoření instalačního souboru aplikace.

---

<sup>1</sup>Více informací zde: <https://www.pythonguis.com/tutorials/packaging-pyqt6-app...>

Dalším užitečným souborem, který byl vytvořen v kořenovém adresáři aplikace, je `.spec` soubor obsahující konfigurace a instrukce, které využívá PyInstaller při stavování výše popsaných dvou složek. PyInstaller lze totiž také spustit příkazem:

```
pyinstaller <hlavni-soubor-aplikace>.spec
```

Ten již využívá zmíněný `.spec` soubor, který je možné si přizpůsobit dle svých potřeb. Typickými příklady úprav `.spec` souboru může být nastavení cest ke zdrojovým adresářům (obsahující ikony, zdrojové kódy oken (*widgets*) aplikace ad.), nastavení ikony aplikace či názvu instalačního souboru nebo zamezení otvírání konzolového okna při spuštění aplikace. Zmíněné nastavení cest (`pathex`) a přidání zdrojových adresářů (`datas`) může vypadat následovně:

Výpis 4.1: PyInstaller – konfigurační soubor `dp-qtdesktopapp.spec`

```
6 add_paths = [  
7   'C:/Git/DP/DP-QtDesktopApp/dp-qtdesktopapp/icons/',  
8   'C:/Git/DP/DP-QtDesktopApp/dp-qtdesktopapp/include/UIs/'  
9 ]  
10  
11 add_data = [  
12   ('dp-qtdesktopapp/icons', 'icons'),  
13   ('dp-qtdesktopapp/include/UIs/*.py', 'UIs')  
14 ]  
15  
16 a = Analysis(  
17   ['dp-qtdesktopapp\\dp-qtdesktopapp.py'],  
18   pathex=add_paths,  
19   binaries=[],  
20   datas=add_data,  
21   ...
```

#### 4.1.4 Poetry

Poetry je velmi sofistikovaný nástroj k efektivní správě závislostí (*dependencies*) pro Python v rámci konkrétního projektu využívající tzv. virtuální prostředí. Virtuální prostředí si lze představit jako zapouzdření či odizolování projektu od okolí. Ve skutečnosti vznikne uvnitř kořenového adresáře projektu složka `.venv`, ve které je nainstalován Python a všechny potřebné balíčky (závislosti). Díky tomuto přístupu je možné využívat v rámci jednoho projektu několik verzí Pythonu zároveň.

#### Výpis 4.2: Poetry – konfigurační soubor `pyproject.toml`

```
1 [tool.poetry]
2 name = "dp-qtdesktopapp"
3 version = "0.1.0"
4 description = "The service desktop app for the test station"
5 authors = ["David Stejskal <211177@vut.cz>"]
6 readme = "README.md"
7 repository = "https://github.com/Dejv99/DP-QtDesktopApp"
8
9 [tool.poetry.dependencies]
10 python = "^3.10"
11 PyQt6 = "^6.4.2"
12 pyqtgraph = "^0.13.2"
13
14 [tool.poetry.group.dev.dependencies]
15 black = "^23.3.0"
16
17 [tool.black]
18 line-length = 88
19
20 [build-system]
21 requires = ["poetry-core"]
22 build-backend = "poetry.core.masonry.api"
```

Vše související s nastavením projektu se dá velmi jednoduše nadefinovat v souboru `pyproject.toml` (ukázka viz Výpis 4.2) uvnitř kořenového adresáře projektu. Závislosti lze rozdělit do skupin podle toho, zda jsou potřeba při běhu výsledného programu, či pouze k vývoji, testování, tvorbě dokumentace, atd. Skupina `[tool.poetry.dependencies]` spadá do implicitní hlavní skupiny, v níž se nachází závislosti, které budou nutné při samotném běhu programu. Oproti tomu skupina `[tool.poetry.group.dev.dependencies]` definuje obsahuje závislosti využívané pouze při vývoji.

S každou změnou souboru `pyproject.toml` dojde také k automatické úpravě souboru `poetry.lock`, který zabraňuje automatickému aktualizování balíčků, a je tak docíleno toho, že projekt využívá přesně specifikovaných verzí závislostí.

U přístupu využívající tzv. globálních balíčků se může při vývoji velmi snadno stát, že některé projekty vyžadují jinou verzi balíčku než ostatní projekty a tak může docházet ke kolizím. Je možné mít globálně nainstalováno současně více verzí balíčku, avšak tento přístup vede dříve nebo později k velmi špatné kontrolovatelnosti.

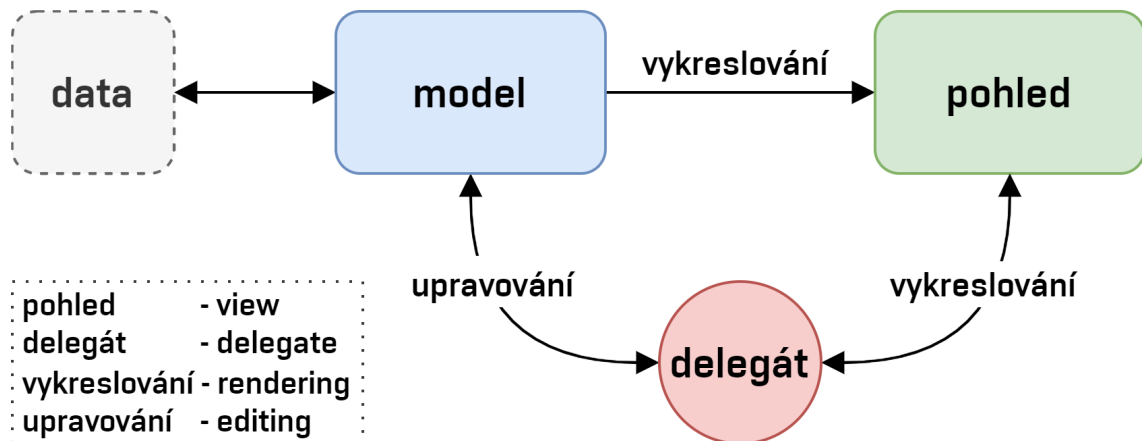
## 4.1.5 PostgreSQL

Jak už podkapitola 4.1 napovídá, v rámci desktopové aplikace pro testovací stanici byla implementována obsluha SQL databáze (viz Cíle práce). Byl zvolen, mezi vývojáři oblíbený, systém pro správu relačních databází PostgreSQL (známý také pod aliasem Postgres). Jedná se o výkonný objektově-relační databázový systém, který je volně dostupný a na němž se vývojáři aktivně podílí již více než 35 let. Díky tomu je známý svou spolehlivostí, robustností a výkonností [23].

PostgreSQL zajišťuje transakce s vlastnostmi *atomicita, konzistence, izolace, trvanlivost* – *Atomicity, Consistency, Isolation, Durability* (ACID), dále automaticky aktualizovatelné pohledy či materializované pohledy, komplexní požadavky, spouštěče, cizí klíče i uchované procedury. Je navržen tak, aby podporoval obsáhlou řadu úloh, od jednoúčelových strojů po webové služby s velkým množstvím souběžných uživatelů nebo datové sklady (*data warehouses*) [20].

## 4.2 Model-View návrhový vzor

Model-View je velmi často používaný návrhový vzor při tvorbě uživatelských rozhraní. Jeho podstata spočívá v tom, že, jednoduše řečeno, odděluje surová data od interakce s nimi a od způsobu jejich zobrazení. Tento přístup umožňuje, že konkrétní data mohou být zobrazena několika různými způsoby, aniž by bylo zasahováno do zdrojových dat. Pro snadnější pochopení následujících odstavců je na Obr. 4.2 znázorněn přehledový diagram kooperace jednotlivých komponent.



Obr. 4.2: Model-View návrhový vzor – přehledový diagram  
(Převzato a upraveno z: <https://doc.qt.io/qt-6/images/modelview-overview.png>)

Součástí tohoto návrhového vzoru je tzv. *Controller*, který definuje, jak se bude chovat uživatelské rozhraní na základě akcí provedených uživatelem.

Se samotnými daty aplikace komunikuje skrze tzv. **Model**, který zprostředkovává rozhraní pro ostatní komponenty architektury. Povaha takové komunikace se odvíjí od typu zdroje dat a způsobu implementace modelu. Model si lze představit jako objekt konkrétní aplikace.

Dalším prvkem této architektury je tzv. **View** (pohled), a ten má na starost vizuální reprezentaci modelu. Od modelu získává modelové indexy, které představují odkazy (reference) na již konkrétní položky dat. S využitím těchto indexů tedy může pohled přistupovat skrze model ke konkrétním elementům dat přímo k jejich zdroji. Jeden model může být zobrazen pomocí více pohledů.

V rámci Qt frameworku byl zaveden ještě tzv. **Delegate** (delegát), který zajišťuje flexibilní zpracování uživatelského vstupu a umožňuje upravovat způsob vykreslování (*rendering*) a upravování (*editing*) položek dat. Při úpravách dat delegát komunikuje s modelem přímo skrze indexy modelu.

Modely, pohledy a delegáti spolu komunikují pomocí tzv. **signals** (signály) a **slots** (sloty). Signály od modelu informují pohledy o změnách na datech uchovávaných zdrojem dat, signály z pohledů poskytují delegátovi informaci o případných uživatelských akcích provedených na konkrétních zobrazených položkách a signály od delegáta poskytují modelu a zobrazením informaci o aktuálním stavu při upravování či zobrazování.

Při vývoji desktopové aplikace bylo využito návrhového vzoru Model-View například při zprovoznování manipulace s SQL databází. Jako model byl z nabídky vybrán  `QSqlQueryModel` , který byl vývojáři Qt Group vyvinut právě za účelem práce s databázemi, aby byl zajištěn určitý komfort vývojářů a především byla zaručena rychlost a spolehlivost daných modelů.

Jako pohled pro vybraný model byl zvolen  `QTableView` , jelikož očekávaný výstup po vykonání například  `SELECT`  dotazu (*query*) je právě tabulka (či sloupec/buňka). Qt nabízí i  `QSqlTableModel`  nebo dokonce  `QSqlRelationalTableModel`  (podporující propojení několika tabulek s využitím cizích klíčů (*foreign keys*)), avšak pro tuto potřebu bohatě postačí „obyčejný“  `QTableView` . V následujícím výpisu 4.3 je názorná ukázka použití předdefinovaného modelu a pohledu.

Výpis 4.3: Qt Model-View –  `QSqlQueryModel` ,  `QTableView`

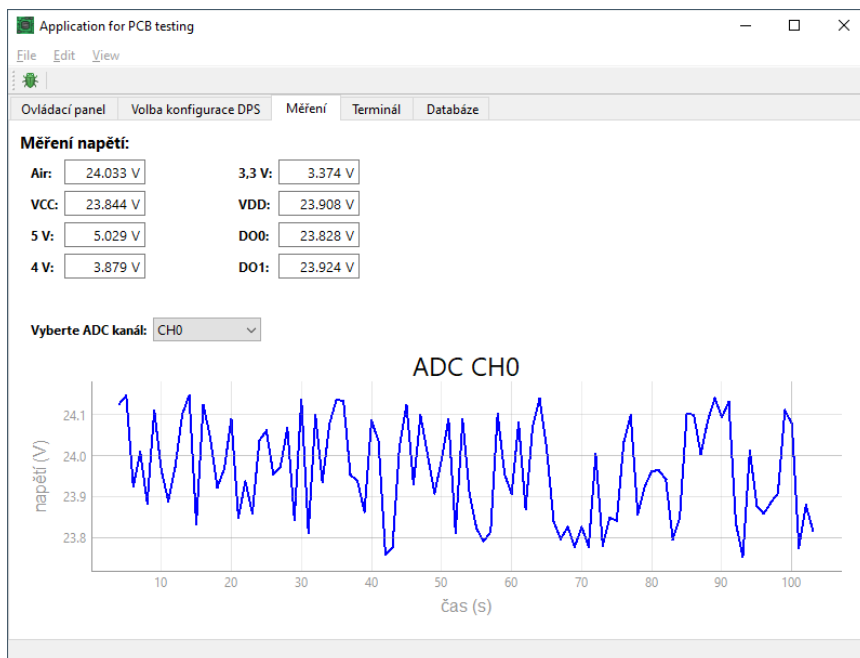
```
1 self.SQLTableView = QtWidgets.QTableView(parent=DatabaseTab)
2 ...
3 self.SQLQueryModel = PyQt6.QtSql.QSqlQueryModel()
4 self.SQLTableView.setModel(self.SQLQueryModel)
```

V této podkapitole bylo čerpáno z [15] a oficiální Qt dokumentace (verze 6.5)<sup>2</sup>.

<sup>2</sup>Oficiální Qt dokumentace: <https://doc.qt.io/qt.html>

## 4.3 Okna obslužné aplikace

Tato podkapitola slouží k prezentaci návrhu první verze obslužné aplikace. Níže jsou ukázána následující okna: Měření (viz Obr. 4.3) a Terminál (viz Obr. 4.4). Okna Ovládací panel, Volba konfigurace DPS a Databáze jsou umístěna do Přílohy D.



Obr. 4.3: Obslužná aplikace – Měření

The screenshot shows the 'Terminál' (Terminal) window of the 'Application for PCB testing' software. The window has a menu bar with 'File', 'Edit', and 'View'. Below the menu bar is a toolbar with a green bug icon. The main area contains a tabbed interface with tabs for 'Ovládací panel', 'Volba konfigurace DPS', 'Měření', 'Terminál', and 'Databáze'. The 'Terminál' tab is active, displaying a section titled 'Možnosti ovládání sériové koinikace přes RS-485' (Options for controlling serial communication over RS-485). Below the title is a dropdown menu for 'Vyberte sériovou linku k připojení' (Select serial port for connection) set to 'COM1'. Below this is a section titled 'Konfigurace sériové linky' (Serial port configuration) with several input fields and dropdown menus: 'Rychlost (baud)' (Baud rate) set to 9600, 'Počet datových bitů' (Number of data bits) set to 8, 'Počet stop bitů' (Number of stop bits) set to 1, 'Parita' (Parity) set to 'Žádná' (None), and 'Řízení toku' (Flow control) set to 'Žádné' (None). A 'Potvrdit' (Confirm) button is located at the bottom right of the configuration section.

Obr. 4.4: Obslužná aplikace – Terminál pro RS-485

## 5 Testovací scénář

Testovací scénář (*test scenario*) představuje jednu ze základních částí testovací dokumentace. Ta by měla být nedílnou součástí procesu testování a vytvářena již při plánování projektu – v tomto případě už při vývoji **testované** DPS. Tester je ten, který nese zodpovědnost za chybovost a garantuje absenci chyb, a právě testovací dokumentace slouží jako jeho opěrný bod [2].

Díky kvalitní definici vstupních podmínek ke spuštění jednotlivých testů a zároveň záznamu o jejich přesném průběhu lze eliminovat množství chyb, které nejsou testováním zachyceny. A co více, slouží zároveň jako výstupní protokol vypovídající o tom, za jakých podmínek byly které testy provedeny. Testovací scénáře tak umožňují efektivně se orientovat v testovacích případech a poskytují lepší představu o tom, které oblasti jsou testy pokryty [2].

### 5.1 Definice a pojmy

Následující sekce popisují základní prvky testovací dokumentace, které představují elementární znalost pro pochopení dané problematiky. Uvádí pojmy jako **testovací případ**, **oblast testování**, **testovací skript**, **metrika hodnocení testování** ad.

#### 5.1.1 Testovací případy

Testovací případ (*test case*) je základní podklad pro testování jednoho specifického místa při konkrétní situaci. Definiuje výchozí podmínky, které musí být splněny za účelem otestovat daný případ a specifikuje typ či formát vstupních dat. Dále by měl jednoznačně udávat podobu očekávaného výsledku [2].

#### 5.1.2 Oblasti testování

Pod pojmem oblast testování si lze představit vlastnost nebo funkci, která musí být otestována v rámci scénáře. Jedná se tedy o logické seskupení konkrétních testovacích případů, které se týkají dané oblasti.

Lze tvrdit, že velmi úzce souvisí s testovacím skriptem (*test script*), který přesně definuje jednotlivé kroky a jejich očekávaný výsledek. Jde o automatický test, jehož podstatou je provést kontrolu konkrétní části (funkce) programu/hardware. Testovací skripty jsou řazeny tak, aby na sebe navazovaly a výstup jednoho skriptu byl tak vstupem dalšího. Příkladem může být situace, kdy pouze v případě funkční sériové komunikace bude možné otestovat digitální vstupy/výstupy (viz sekce 5.2.2). Spojením několika testovacích skriptů vzniká právě testovací scénář [2].



### 5.1.3 Metriky hodnocení testování

Metriky (kritéria) slouží k tomu, aby bylo možné určit, zda scénář proběhl úspěšně. Existuje nepřeberné množství metrik, avšak pro každý testovací případ je nutné vybrat ty, které nejefektivněji a nejvěrohodněji otestují konkrétní vlastnost [14].

Při testování DPS mají velkou vypovídající hodnotu především metriky elektrického a funkčního testování. Mezi nejčastěji používané **metriky elektrického testování** patří například měření hodnot elektrického napětí, proudu, odporu, kvality signálové integrity („celistvosti“) či zkraslení ad. Za **metriky funkčního testování**, které se používají k posouzení funkčnosti/výkonnosti DPS během různých funkčních scénářů, lze považovat například spotřebu energie, rychlost přenosu dat a kompatibilitu (srovnání desek sloužící stejnému účelu ale s využitím jiných komponent). Metriky elektrického i funkčního testování spolu pochopitelně velmi úzce souvisí. Dalšími více obecnými metrikami mohou být:

- testovací pokrytí (*test coverage*) – určuje procentuálně, jak velká část programu/požadavků byla otestována,
- hustota defektů (*defect density*) – měří počet defektů nalezených v konkrétních testovacích případech a slouží tak k identifikaci oblastí, které je potřeba důkladněji otestovat,
- doba provedení testu (*test execution time*) – může být nápomocen při hledání oblastí, jejichž testování trvá nadměrné množství času a mohlo by tak vypovídat o výkonnosti programu,
- poměr úspěšný/neúspěšný test (*test pass/fail ratio*) – určuje procentuální poměr mezi testy co prošly a neprošly, a může tak vypovídat o celkové kvalitě testovaného programu,
- *Mean Time Between Failures* (MTBF) – měří průměrnou dobu mezi selháními programu resp. jeho stabilitu,
- *Mean Time To Failures* (MTTF) – udává průměrný čas, za který dojde k selhání programu a může tak pomoci určit, zda je program dostatečně spolehlivý.

## 5.2 Tvorba testovacího scénáře pro testovanou DPS

V rámci této podkapitoly je podrobně popsán a znázorněn postup při tvorbě testovacího scénáře pro testovanou DPS. Nejprve jsou vydefinovány oblasti testování, následně konkrétní případy, na jejichž základě jsou sestaveny testovací skripty. Z těch je na závěr sestaven finální testovací scénář.

## 5.2.1 Oblasti testování

K hrubému rozdělení testovacího scénáře do logických úseků je nejprve provedeno vydefinování oblastí testování. To je znázorněno pomocí následujícího seznamu:

- načtení konfigurace,
- komunikace po RS-485 – mezi obslužnou aplikací a testovanou DPS,
- měření napětí na digitálních výstupech testované DPS a měření napájecích napětí,
- ovládání digitálních vstupů.

## 5.2.2 Případy testování

Když už jsou vydefinované oblasti testování, je nutné se zamyslet ideálně nad všemi možnými případy, které by mohly nastat – čím více je jich namyšleno, tím většího testovacího pokrytí bude dosaženo (viz sekce 5.1.3).

Spojení mezi počítačem a testovací DPS by mělo být provedeno neprodleně po připojení USB kabelu. Pakliže se tak nestalo, je doporučeno ověřit, zda proběhlo nahrání firmwaru do testovací DPS úspěšně. Po úspěšném připojení je již možné nahrát konfiguraci přizpůsobovací DPS. To je prováděno v okně „Volba konfigurace DPS“ obslužné aplikace (viz Příloha D.2).

### Načtení konfigurace přizpůsobovací DPS

Dříve nahranou konfiguraci přizpůsobovací DPS je po spuštění automatického testu nutné načíst. Načtená data rozhodnou o tom, který testovací scénář bude zvolen a také budou důležitou součástí vygenerovaného finálního reportu automatického testu. Po obdržení příkazu, poslaném po RS-485 z obslužné aplikace, je mikroprocesorem na testovací DPS po sběrnici I2C provedeno samotné načtení dat. Mohou nastat následující případy:

- načtená data nic neobsahují – konfigurace přizpůsobovací DPS pravděpodobně nebyla (úspěšně) nahrána,
- data úspěšně načtena – test proběhl v pořádku, lze postoupit k dalšímu kroku testovacího scénáře.

### Komunikace po RS-485

Pro otestování funkce RS-485 (spojení mezi obslužnou aplikací a testovanou DPS) je nejprve nutné komunikaci správně nakonfigurovat (komunikační port, rychlost komunikace, počet datových a stop bitů, parita, příp. řízení toku) v obslužné aplikaci (viz Obr. 4.4) – nutno upozornit, že součástí okna pro terminál zatím není oblast pro zobrazování odeslané a přijaté zprávy, tzn. aktuálně slouží pouze ke konfiguraci.

Nejjednodušší možností je poslat předem definovanou zprávu a očekávat konkrétní odpověď. Při tomto přístupu lze očekávat 3 možné varianty reakcí na odeslanou zprávu:

- odpověď neobdržena (v určitém časovém rozmezí) – nejspíše chyba v zapojení (příp. špatný kontakt) či nesprávně zvolen komunikační port,
- obdržena neočekávaná odpověď – nejspíš špatně nastavená rychlost komunikace,
- obdržena očekávaná odpověď – test proběhl v pořádku.

### **Měření napětí pomocí 8-kanálového ADC**

Aby bylo možné měřit napětí na digitálních výstupech testované DPS (viz Air, DO0, DO1 na Obr. 2.14), musí být nejprve poslán příkaz po RS-485 pro aktivaci těchto výstupů. Mohou nastat následující případy:

- odpověď neobdržena (v určitém časovém rozmezí) – nelze pokračovat v měření napětí na digitálních výstupech,
- obdržena očekávaná odpověď – test proběhl v pořádku, lze zahájit měření napětí na digitálních výstupech.

Po obdržení očekávané odpovědi o provedené aktivaci digitálních výstupů je zahájeno měření napětí na všech kanálech A/D převodníku (tzn. včetně napájecích napětí na ostatních testovacích pinech). V opačném případě budou změřena pouze napájecí napětí.

### **Ovládání digitálních vstupů**

V případě digitálních vstupů (viz DI0 až DI5 na Obr. 2.14) je potřeba nejdříve aktivovat požadované porty GPIO expandéru. Tímto by měly být aktivovány konkrétní vstupy mikroprocesoru na testované DPS. Následně je poslán příkaz za účelem vyžádání odezvy obsahující stavy (vysoká či nízká logická úroveň) všech těchto vstupů. Mohou nastat následující případy:

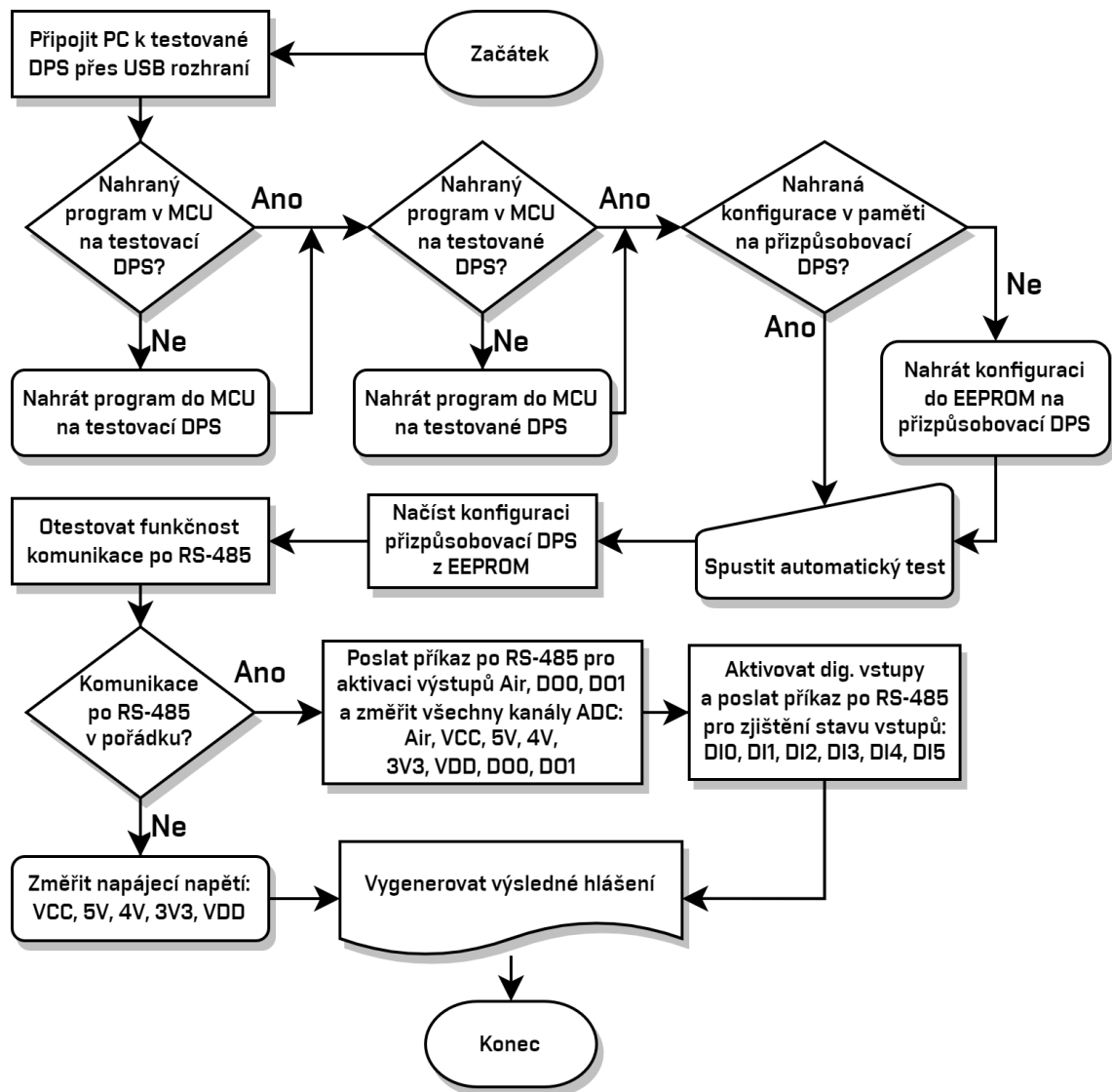
- odpověď neobdržena (v určitém časovém rozmezí) – nelze vyhodnotit stavy digitálních vstupů,
- obdržena očekávaná odpověď – test proběhl v pořádku, stavy digitálních vstupů byly obdrženy a zaznamenány.

## **5.2.3 Finální testovací scénář**

Ještě před tím než bude možné spustit automatický test, je nutné provést určité prerekvizity testování. Tou první a nejdůležitější je nahrání samotného firmwaru

do mikroprocesoru SAM D21 na testovací DPS. Obdobně je nutné nahrát firm-ware do MCU na testované DPS, ve kterém budou naprogramované konkrétní akce na příkazy posílané z obslužné aplikace po RS-485. Následuje nahrání konfigurace přizpůsobovací DPS do její EEPROM.

Po spuštění automatického testu se nejprve načte konfigurace DPS z EEPROM. Následuje otestování komunikace po RS-485 posláním předdefinované zprávy s očekáváním konkrétní odpovědi. V případě negativního výsledku funkčnosti se změní pouze napájecí napětí VCC, 5V, 4V, 3V3, VDD a vygeneruje se výsledné hlášení.



Obr. 5.1: Vývojový diagram testovacího scénáře

V opačném případě je poslán příkaz pro aktivaci dig. výstupů Air, DO0, DO1 a následně jsou změřeny všechny kanály ADC. Potom jsou aktivovány dig. vstupy DI0, DI1, DI2, DI3, DI4, DI5 (bráno vzhledem k testované DPS) a poslán příkaz pro zjištění stavu těchto vstupů (vyhodnoceno MCU na testované DPS). V tomto

okamžiku je vše úspěšně otestováno a lze vygenerovat kompletní výslednou zprávu o testu. Vývojový diagram popsaného testovacího scénáře lze vidět na Obr. 5.1. Ve Výpisu 5.1 je ukázán příklad takové výsledné zprávy.

Výpis 5.1: Výsledné hlášení o provedeném automatickém testu

```
===== Final Automatic Test Report =====
Date: 2023-05-08
Time: 08:58:08

General Information:
- DPS Name: Prizpusobovaci DPS pro Aromatizacni system
- Author Name: David Stejskal
- DPS Version: v1.0
- DPS Configuration Date: 08/05/2023
- DPS Description: Tato Prizpusobovaci DPS je testovacim podstavcem
  pro ridici desku Aromatizacniho systemu.

Test Configuration:
- ADC Channels: 8
- GPIO Channels: 8
- Dig. Inputs: 6
- Dig. Outputs: 2
- RS-485, I2C

Test Results:
- Test Points:
  - VCC: 21.913 V — PASS
  - Air: 0.000 V — FAIL
  - +5V: 4.764 V — PASS
  - +4V: 0.000 V — FAIL
  - +3V3: 3.293 V — PASS
  - VDD: 0.000 V — FAIL
  - DO0: 0.000 V — FAIL
  - DO1: 0.000 V — FAIL

Pass/Fail Criteria:
- Voltage in range +/-10.00 %

Conclusion:
The following TPs are not within the acceptable range:
- Air, +4V, VDD, DO0, DO1
```

V Příloze E jsou zobrazeny následující ukázky kódu: Hlavní testovací skript, Načtení názvu DPS z její konfigurace v EEPROM, Změření všech kanálů ADC, Aktivace dig. vstupů pro testovanou DPS.

# Závěr

V rámci diplomové práce byl vytvořen celkový návrh a následně elektrické, hardwarové i programové řešení testovacího systému vyrobených DPS obsahující testovací piny pro výstupní stanoviště výroby. Jedním ze zásadních přínosů této práce je samotná univerzálnost testovací desky, která dokáže, mimo komunikaci s desktopovou aplikací za účelem provádění a vizualizace automatických testů, programovat i ladit různé typy mikroprocesorů. Deska také disponuje poměrně velkým množstvím prvků umožňujících výraznou variabilitu při programování i ladění. Přínosem přizpůsobovací desky je především její obrovská užitečnost pro zautomatizování testování vyrobených DPS, čímž je tento proces výrazně zjednodušen a urychlen. Vytvořený testovací scénář přesně definuje konkrétní případy testování a pomocí sestaveného vývojového diagramu je tak mnohem snazší se v procesu testování zorientovat, případně identifikovat problém, který v průběhu tohoto procesu nastal.

Kapitola 1 byla zaměřena na bázi teoretických znalostí potřebných pro lepší pochopení souvislostí, jimiž se zabývá tato diplomová práce. Z důvodu komplexnosti práce jsou součástí teoretického úvodu pouze některé oblasti a předpokládá se, že čtenář již má jisté znalosti z této problematiky. Byl zde podrobně popsán výrobní proces desek plošných spojů a znázorněno spektrum vad, které během tohoto procesu vznikají. Následuje poměrně rozsáhlá podkapitola 1.3 o samotném pojmu testování. Kapitola je zakončena podkapitolou 1.4 zabývající se standardy pro sériovou komunikaci.

V rámci kapitoly 2, pojednávající podrobně o návrhu přizpůsobovací a testovací DPS, byla vytvořena bloková schémata (viz Obr. 2.2 a Obr. 2.24) a souhrnné vysvětlující tabulky nejpodstatnějších použitých komponent (viz Tab. 2.2 a Tab. 2.3). V návaznosti na to byly prezentovány a popsány návrhy elektrických schémat a následně realizovány jim odpovídající desky plošných spojů (viz Obr. 2.12 a Obr. 2.29). Pro lepší představu byla na Obr. 2.1 znázorněna vzájemná konfigurace přizpůsobovací, testovací i testované DPS. Závěrem (podkapitola 2.4) je prezentován kompletní testovací systém včetně vsazeného testovaného produktu (viz Obr. 2.34 a Obr. 2.35).

V kapitole 3 byl stručně popsán postup tvorby programového řešení testovacího zařízení. Byly zde uvedeny důležité poznatky, které bylo nutné znát před samotnou implementací ovladačů použitých periférií v jazyce C. V rámci jednotlivých sekcí byly prezentovány krátké výpisy kódu obsahující deklarace implementovaných funkcí konkrétních periférií (viz Výpisy 3.1, 3.2 a 3.3). Po implementaci ovladačů potřebných I2C periférií následovala stěžejní část – zprovoznění sériové komunikace s počítačovou aplikací. Postupně tvořený firmware byl průběžně laděn na oficiální vývojové desce SAMC21 Xplained PRO od společnosti Microchip, která disponuje vestavěným rozhraním eDBG pro snadné programování a ladění po rozhraní USB.

Při ladění I2C periferií byl využit osciloskop RIGOL DS1054Z, konkrétně jeho funkce pro dekódování sériové komunikace po signálech SDA a SCL. Následně byl firmware implementován na MCU SAMD21 osazený na testovací DPS.

Kapitola 4 dává čtenáři vhled do procesu tvorby počítačové aplikace v programovacím jazyce Python. Je zde uvedeno, jakých programových nástrojů lze využít nejen k vývoji takové aplikace, ale i k následné uživatelsky přívětivé distribuci. V podkapitole 4.2 byl popsán návrhový vzor *Model-View* a v podkapitole 4.3 byla ukázána některá okna počítačové aplikace (viz Obr. 4.3 a Obr. 4.4, ostatní v Příloze D).

Poslední kapitola 5 uvádí do problematiky tvorby testovacího scénáře. Na úvod byly vysvětleny základní pojmy a následoval návrh testovacího scénáře pro konkrétní případ testované DPS (viz podkapitola 5.2). Závěrem byl prezentován odpovídající vývojový diagram (*flow chart*, viz Obr. 5.1), vygenerovaná výsledná testovací zpráva (viz Výpis 5.1) a ukázky testovacích skriptů (viz Příloha E).

Z důvodu komplexnosti této práce byl způsob konfigurace po rozhraní Bluetooth přesunut do případné další iterace tohoto zařízení. Práce s databázovým systémem nebyla nakonec po dohodě s vedoucím zprovozněna, avšak byly provedeny potřebné kroky pro porozumění dané problematice včetně pokusu o implementaci do desktopové aplikace (viz Příloha D.3).

# Literatura

1. ALTERA CORPORATION. *Designing Power Isolation Filters with Ferrite Beads for Altera FPGAs* [online]. 2009-07 [cit. 2022-12-29]. Tech. zpr., AN-583-1.0. Dostupné z: [https://www.altera.com/content/dam/altera-www/global/en\\_US/pdfs/literature/an/an583.pdf](https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/an/an583.pdf).
2. ANON. *Testovací dokumentace - plán, scénář, případ* [online]. bez data [cit. 2023-05-07]. Dostupné z: [http://test.swtestovani.cz/index.php?option=com\\_content&view=article&id=15:testovaci-dokumentace-plan-scenar-pipad&catid=3:zaklady&Itemid=11](http://test.swtestovani.cz/index.php?option=com_content&view=article&id=15:testovaci-dokumentace-plan-scenar-pipad&catid=3:zaklady&Itemid=11).
3. ECO, Jefferson; LIMJOCO, Aldrick. *Ferrite Bead Demystified* [online]. 2015-08 [cit. 2022-12-29]. Tech. zpr., AN-1368. Dostupné z: <https://www.analog.com/media/en/analog-dialogue/volume-50/number-1/articles/ferrite-beads-demystified.pdf>.
4. ELINUX. *RPi GPIO Interface Circuits* [online]. 2023-02 [cit. 2023-05-03]. Dostupné z: [https://elinux.org/RPi\\_GPIO\\_Interface\\_Circuits#Level\\_Shifters](https://elinux.org/RPi_GPIO_Interface_Circuits#Level_Shifters).
5. EMA DESIGN AUTOMATION. *Design for Test (DFT) - What PCB Design Engineers Need to Know* [online]. 2020-12 [cit. 2022-10-18]. Dostupné z: <https://youtu.be/ehtSgJQ3oKA>.
6. KUPHALDT, Tony R. *EIA/TIA-232, 422, and 485 Networks: Chapter 6 – Digital Data Acquisition and Networks in Control Systems* [online]. 2019-12 [cit. 2022-11-15]. Dostupné z: <https://control.com/textbook/digital-data-acquisition-and-networks/eiatia-232-422-and-485-networks/>.
7. MAXIM, Dallas Semiconductor. *Fundamentals of RS-232 Serial Communications* [online]. 2001-03 [cit. 2022-11-15]. Č. AN83. Dostupné z: <https://www.maximintegrated.com/en/design/technical-documents/tutorials/8/83.html>.
8. MAXIM, Dallas Semiconductor. *RS-485 (EIA/TIA-485) Differential Data Transmission System Basics* [online]. 2001-11 [cit. 2022-11-15]. Tech. zpr. Dostupné z: <https://pdfserv.maximintegrated.com/en/an/AN723.pdf>.
9. MICROSOFT. *Python in Visual Studio Code* [online]. 2022-03 [cit. 2023-04-10]. Dostupné z: <https://code.visualstudio.com/docs/languages/python>.
10. MODBUS ORGANIZATION. *MODBUS over serial line specification and implementation guide V1.02* [online]. 2006-12 [cit. 2022-12-31]. Dostupné z: [www.modbus.org/docs/Modbus\\_over\\_serial\\_line\\_V1\\_02.pdf](http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf).



11. MULLINS, Edward; WILLIAMS, Ian; LO VOI, Giuseppe; PUZIO, Raphael. *How to Properly Configure Unused Operational Amplifiers* [online]. 2018-09 [cit. 2023-05-03]. Tech. zpr., SBOA204A. Dostupné z: <https://www.ti.com/lit/pdf/sboa204?keyMatch=SBOA204A>.
12. ORESJO, Stig. Features-One Billion Solder Joints... and Counting-One of the most comprehensive solder joint studies to date reveals that the industry is a long way from 100 ppm defect levels. *Circuits Assembly*. 2001, roč. 12, č. 2, s. 26–32.
13. PCBCART. *Printed Circuit Boards Assembly (PCBA) Process* [online]. 2017-03 [cit. 2022-11-19]. Dostupné z: <https://www.pcbcarts.com/article/content/pcb-assembly-process.html>.
14. PETERSON, Zachariah. *PCB Testing 101: Important Methods and Metrics* [online]. 2021-11 [cit. 2023-05-07]. Dostupné z: <https://resources.altium.com/p/pcb-testing-101-important-methods-and-metrics>.
15. QT COMPANY. *Model/View Programming* [online]. 2023-04 [cit. 2023-04-11]. Dostupné z: <https://doc.qt.io/qt-6/model-view-programming.html>.
16. RAYMING TECHNOLOGY. *Basic Thing You Should Know About PCB Assembly Process* [online]. 2022-11 [cit. 2022-11-19]. Dostupné z: <https://www.raypcb.com/printed-circuit-boards-assembly-process/>.
17. RENESAS. *RS-422 vs RS-485 – Similarities and Key Differences* [online]. 2017-12 [cit. 2022-11-15]. Tech. zpr., AN1989. Dostupné z: <https://www.renesas.com/us/en/document/apn/an1989-rs-422-vs-rs-485-similarities-and-key-differences>.
18. SWAGATAM. *Voltage Regulator Circuits using Transistor and Zener Diode* [online]. 2022-04 [cit. 2022-12-29]. Dostupné z: [https://www.homemade-circuits.com/simple-voltage-regulator-circuits-using-transistor-and-zener-diode/#Series\\_Transistor\\_Regulator\\_Circuit](https://www.homemade-circuits.com/simple-voltage-regulator-circuits-using-transistor-and-zener-diode/#Series_Transistor_Regulator_Circuit).
19. TEXAS INSTRUMENTS. *IEC 61000-4-x Tests for TI's Protection Devices* [online]. 2015-06 [cit. 2022-12-27]. Tech. zpr., SLVA711. Dostupné z: <https://www.ti.com/lit/pdf/slva711>.
20. THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL Documentation: What Is PostgreSQL?* [Online]. 2023-02 [cit. 2023-04-28]. Dostupné z: <https://www.postgresql.org/docs/current/intro-what-is.html>.

21. UNGAR, Louis Y. Design for testability (DFT) to overcome functional board test complexities in manufacturing test. *Proc. IPC APEX* [online]. 2017, s. 1–40 [cit. 2022-10-18]. Dostupné z: <https://smtnet.com/library/files/upload/dft-functional-test.pdf>.
22. VRBA, Kamil; HANÁK, Pavel. *Konstrukce elektronických zařízení*. 1. vyd. Brno: Vysoké učení technické v Brně, Nakladatelství VUTIUM, 2021. ISBN 978-80-214-5957-1.
23. WIKIPEDIA. *PostgreSQL* [online]. 2023-04 [cit. 2023-04-28]. Dostupné z: <https://en.wikipedia.org/wiki/PostgreSQL>.
24. WIKIPEDIA. *Qt (software)* [online]. 2023-04 [cit. 2023-04-10]. Dostupné z: [https://en.wikipedia.org/wiki/Qt\\_\(software\)](https://en.wikipedia.org/wiki/Qt_(software)).
25. WIKIPEDIA. *USB hub* [online]. 2023-03 [cit. 2023-03-01]. Dostupné z: [https://en.wikipedia.org/wiki/USB\\_hub](https://en.wikipedia.org/wiki/USB_hub).

## Seznam symbolů a zkratek

<b>MCU</b>	Microcontroller Unit
<b>DPS</b>	deska plošných spojů
<b>THT</b>	Through-hole Technology
<b>PCBA</b>	Printed Circuit Board Assembly
<b>SMD</b>	Surface Mount Device
<b>IC</b>	integrovaný obvod – Integrated Circuit
<b>TP</b>	Test Point
<b>AOI</b>	Automated Optical Inspection
<b>AXI</b>	Automated X-Ray Inspection
<b>ICT</b>	In-Circuit Testing
<b>FBT</b>	Functional Board Test
<b>RRP</b>	Risk Reduction Process
<b>RS</b>	Recommended Standard
<b>TIA/EIA</b>	Telecommunications Industry Association/Electronic Industries Alliance
<b>DTE</b>	Data Terminal Equipment
<b>DCE</b>	Data Communications Equipment
<b>UART</b>	Universal Asynchronous Receiver-Transmitter
<b>TVS</b>	Transient Voltage Suppression diode
<b>ESD</b>	Electrostatic Discharge
<b>EFT</b>	Electrical Fast Transient
<b>IEC</b>	International Electrotechnical Commission
<b>MOSFET</b>	Metal Oxide Semiconductor Field Effect Transistor
<b>ZD</b>	Zenerova dioda

<b>USB</b>	Universal Serial Bus
<b>JTAG</b>	Joint Test Action Group
<b>I2C</b>	Inter-Integrated Circuit
<b>UPDI</b>	Unified Program and Debug Interface
<b>FB</b>	Ferrite Bead
<b>DC</b>	Direct Current
<b>RAM</b>	Random Access Memory
<b>ADC</b>	Analog to Digital Converter
<b>DAC</b>	Digital to Analog Converter
<b>PWM</b>	Pulse Width Modulation
<b>DMA</b>	Direct Memory Access
<b>MTT</b>	Multi Transaction Translator
<b>PLC</b>	Programmable Logic Controller
<b>RTU</b>	Remote Terminal Unit
<b>SCADA</b>	Supervisory Control and Data Acquisition
<b>CRC</b>	Cyclic-Redundant Checksums
<b>ASCII</b>	American Standard Code for Information Interchange
<b>ISO/OSI</b>	International Organization for Standardization/Open Systems Interconnection
<b>HMI</b>	Human-Machine Interface
<b>OPC</b>	Open Platform Communications
<b>PDU</b>	Protocol Data Unit
<b>PDI</b>	Program and Debug Interface
<b>EMC</b>	Electromagnetic Compatibility
<b>EEPROM</b>	Electrically Erasable Programmable Read-Only Memory

<b>GPIO</b>	General-Purpose Input/Output
<b>SWD</b>	Serial Wire Debug
<b>ISP</b>	In-circuit Serial Programming
<b>TPI</b>	Tiny Programming Interface
<b>CMOS</b>	Complementary Metal–Oxide–Semiconductor
<b>SQL</b>	Structured Query Language
<b>CRUD</b>	vytvořit, číst, aktualizovat, odstranit – Create, Read, Update, Delete
<b>GUI</b>	Graphical User Interfaces
<b>QML</b>	Qt Modeling Language
<b>JSON</b>	JavaScript Object Notation
<b>XML</b>	Extensible Markup Language
<b>ACID</b>	atomicita, konzistence, izolace, trvanlivost – Atomicity, Consistency, Isolation, Durability
<b>GND</b>	ground
<b>MTBF</b>	Mean Time Between Failures
<b>MTTF</b>	Mean Time To Failures
<b>SE</b>	Single-ended

# Seznam příloh

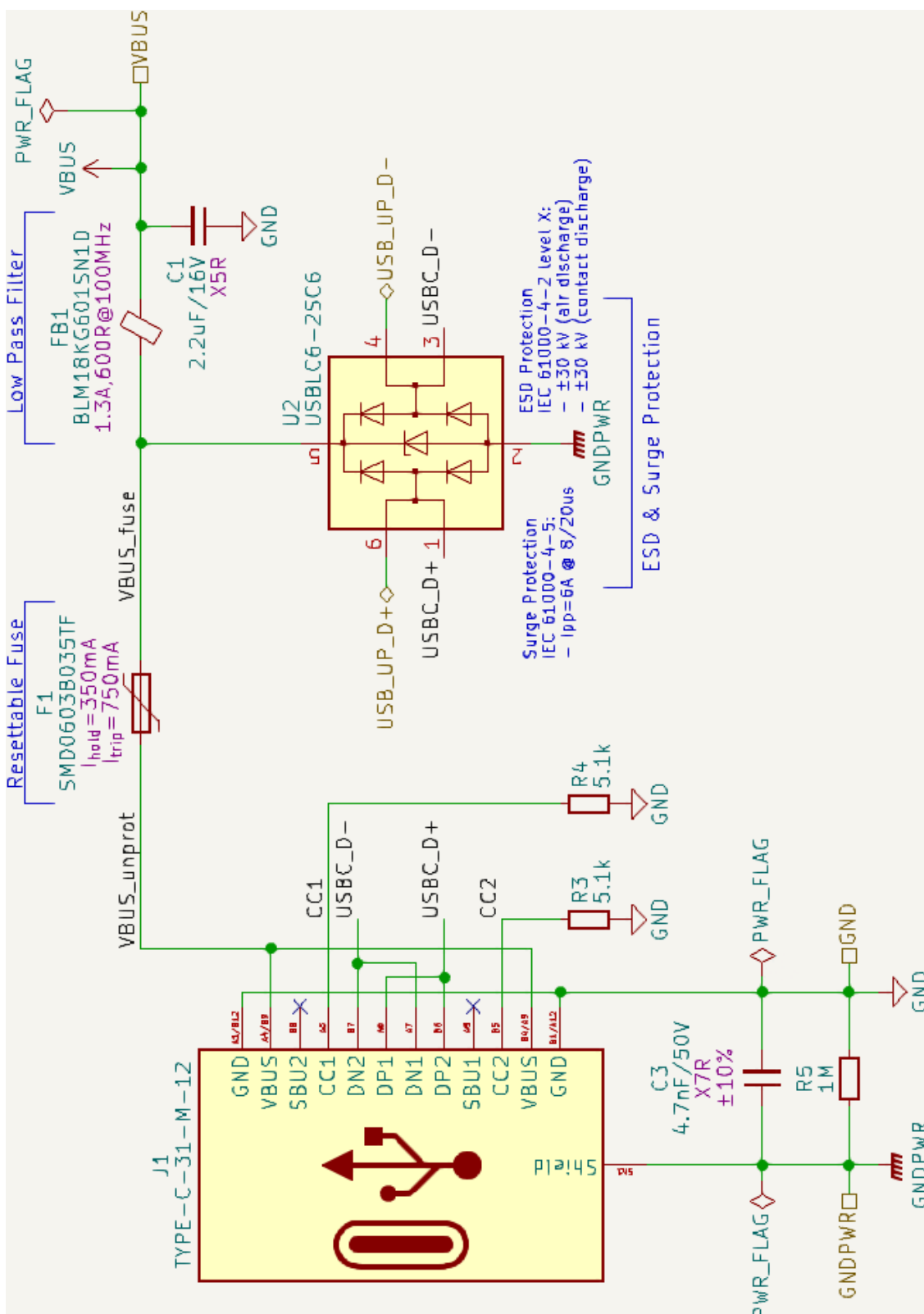
<b>A</b>	<b>Obsah elektronické přílohy</b>	<b>87</b>
<b>B</b>	<b>Elektrická schémata</b>	<b>88</b>
B.1	Jištění a filtrace napájecího obvodu . . . . .	88
B.2	Programátor modulů ESP32 . . . . .	89
B.3	UPDI – ATmega32U4 . . . . .	90
B.4	Převodník USB na 2x UART – čip FT2232HL . . . . .	91
<b>C</b>	<b>Výrobní výkresy desek</b>	<b>92</b>
C.1	Prizpůsobovací DPS – Výrobní výkres s pájecí maskou (vrchní vrstva)	92
C.2	Testovací DPS – Výrobní výkres s pájecí maskou (vrchní vrstva) . .	93
<b>D</b>	<b>Obslužná aplikace</b>	<b>94</b>
D.1	Hlavní okno k ovládní testování . . . . .	94
D.2	Okno pro nastavení/výběr konfigurace . . . . .	95
D.3	Okno pro obsluhu databáze . . . . .	96
<b>E</b>	<b>Ukázky testovacích skriptů</b>	<b>97</b>
E.1	Hlavní testovací skript . . . . .	97
E.2	Načtení názvu DPS z její konfigurace v EEPROM . . . . .	97
E.3	Změření všech kanálů ADC . . . . .	98
E.4	Aktivace dig. vstupů pro testovanou DPS . . . . .	98

# A Obsah elektronické přílohy

- /.....kořenový adresář přiloženého archivu
  - └ A.1 Elektronická verze závěrečné práce
    - └ STEJSKAL-DP-Testing\_Station.pdf
  - └ B.1 Elektrická schémata testovací DPS
    - └ conn\_dsub-37.kicad\_sch
    - └ esp-prog.kicad\_sch
    - └ mcu.kicad\_sch
    - └ quad\_usb\_hub.kicad\_sch
    - └ testing-board-for-pcb-testing.kicad\_sch
    - └ updi.kicad\_sch
    - └ usb-c\_power\_supply.kicad\_sch
    - └ usb\_to\_dual\_rs-232.kicad\_sch
  - └ B.2 Výrobní podklady testovací DPS
    - └ testing-board-for-pcb-testing.kicad\_pcb
  - └ C.1 Elektrická schémata přizpůsobovací DPS
    - └ communication.kicad\_sch
    - └ conn\_dsub-37.kicad\_sch
    - └ customizable-stand-for-pcb-testing.kicad\_sch
    - └ digital-outputs.kicad\_sch
    - └ peripheries.kicad\_sch
    - └ power\_supply.kicad\_sch
  - └ C.2 Výrobní podklady přizpůsobovací DPS
    - └ customizable-stand-for-pcb-testing.kicad\_pcb
  - └ D.1 Firmware testovací stanice
    - └ test-station.....kořenový adresář projektu
    - └ .gitignore.....specifikace záměrně nesledovaných souborů verz. systémem
    - └ README.md.....úvodní informace dokumentace projektu
  - └ E.1 Zdrojové soubory obslužné aplikace
    - └ .gitignore.....specifikace záměrně nesledovaných souborů verz. systémem
    - └ dp-qtdesktopapp.spec.....konfigurační soubor pro InstallForge
    - └ poetry.lock
    - └ pyproject.toml.....konfigurační soubor pro Poetry
    - └ README.md
  - └ E.2 Instalační soubor obslužné aplikace
    - └ InstallForgeSettings.ifp .... nastavení InstallForge pro opakované spuštění
    - └ PCB-test-application.exe.....spustitelný instalační soubor
  - └ F.1 Ostatní soubory
    - └ D-Sub DC-37 Rozhraní.xlsx.....rozhraní a signály mezi test. a přizp. DPS
    - └ EEPROM Registry.xlsx
    - └ Final Test Report.txt ..... výsledná zpráva automatického testu

## B Elektrická schémata

### B.1 Jištění a filtrace napájecího obvodu

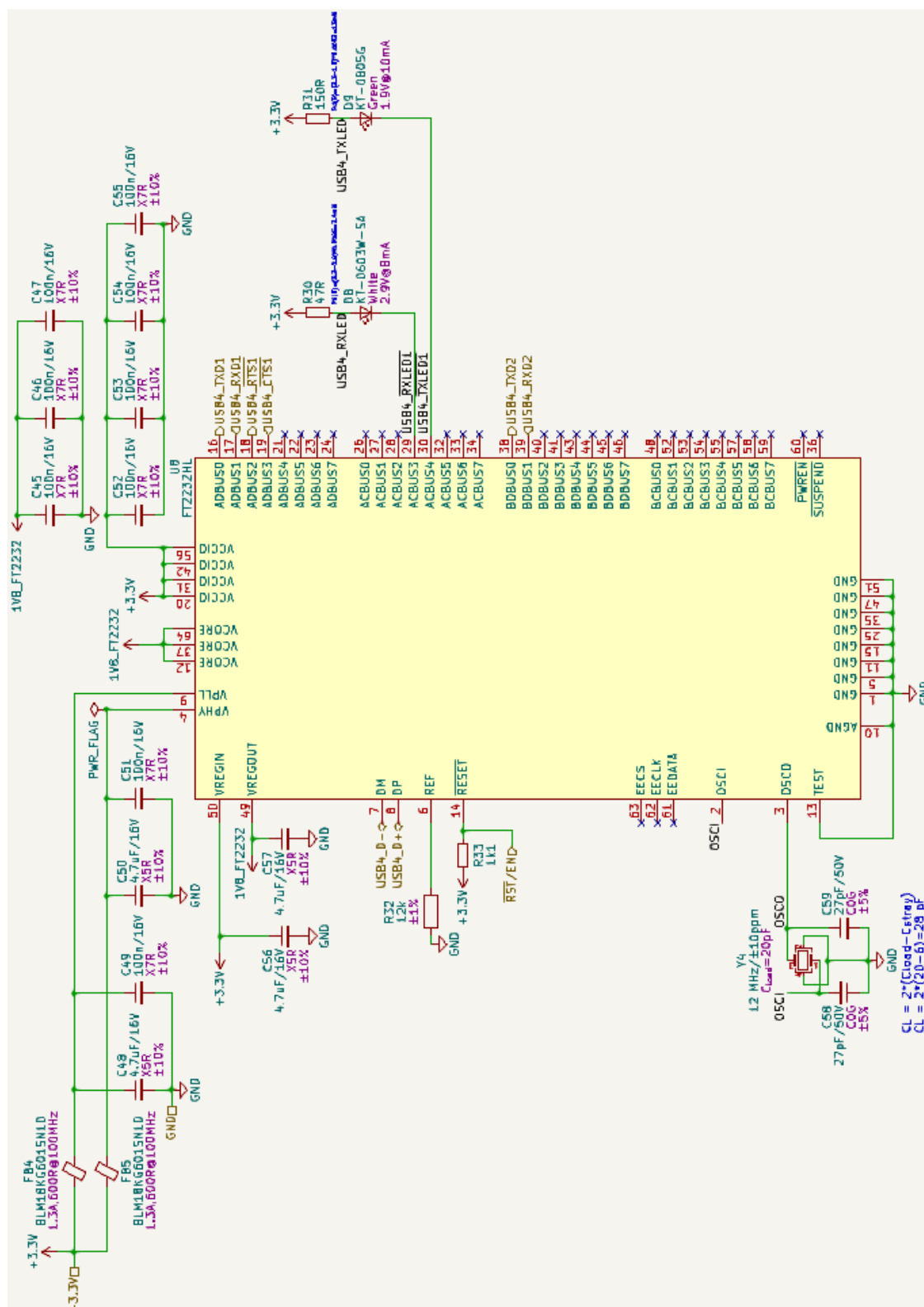








## B.4 Převodník USB na 2x UART – čip FT2232HL

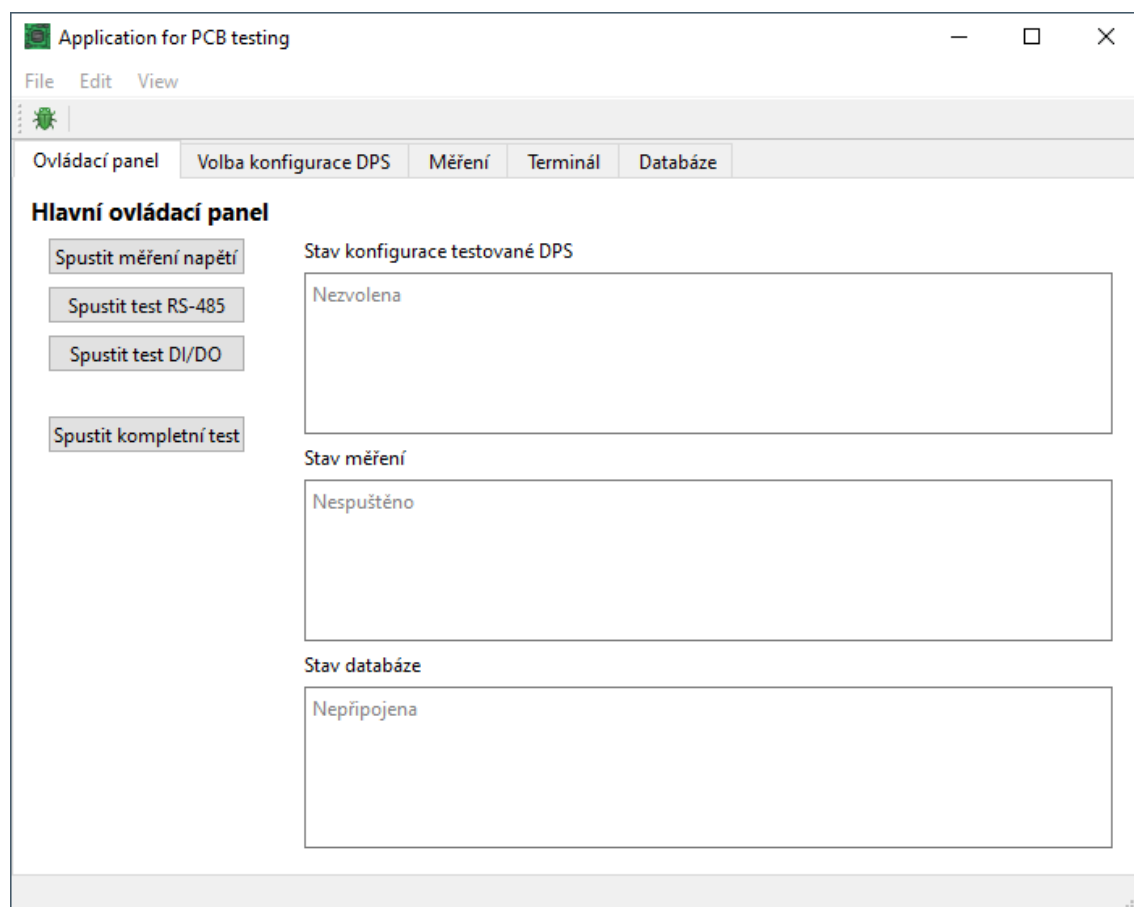






# D Obslužná aplikace

## D.1 Hlavní okno k ovládání testování



## D.2 Okno pro nastavení/výběr konfigurace

Application for PCB testing

File Edit View

Ovládací panel Volba konfigurace DPS Měření Terminál Databáze

**Volba konfigurace DPS:**

Název DPS: Přizpůsobovací DPS pro Aromatizační systém

Jméno autora: David Stejskal

Verze DPS: v1.0

Datum konfigurace: 09. 05. 2023

Popis DPS: Tato přizpůsobovací DPS představuje testovací podstavec pro řídicí desku Aromatizačního systému.

Konfigurace DPS:

Počet digitálních vstupů:	6	Rozhraní:
Počet digitálních výstupů:	2	<input checked="" type="checkbox"/> RS-485
Počet ADC kanálů:	8	<input type="checkbox"/> RS-232
Počet vstupů/výstupů GPIO expandéru:	8	<input type="checkbox"/> CAN
		<input type="checkbox"/> SPI
		<input checked="" type="checkbox"/> I2C
		<input type="checkbox"/> USART

Výchozí hodnoty Potvrdit

## D.3 Okno pro obsluhu databáze

Application for PCB testing

File Edit View

Ovládací panel Volba konfigurace DPS Měření Terminál Databáze

### Připojení a obsluha databáze

**Login**

Uživatel:

Heslo:

Databáze:

**Stav databáze**

Výběr záznamů z tabulky test\_table proběhlo úspěšně!  
Bylo vybráno 11 záznamů.

---

**Obsluha databáze**

**Vytvořit záznam**

Sloupce:

Hodnoty:

**Zobrazit záznamy**

Sloupce:

Podmínka:

Max. limit řádků:

**Aktualizovat záznam**

Sloupce:

Hodnoty:

Podmínka:

**Smazat záznam**

Podmínka:

Zobrazená tabulka:

	id	employee	address	department	birth_date
1	3	Petra	Rymarov	MandelU	01/11/1998
2	2	David	Brno	VUT	04/04/1999
3	4	Frantisek	Brno	HON	01/01/1990
4	5	Petr	Olomouc	VUT	01/03/1998
5	6	Daniel	Nachod	ATAS	14/05/1997
6	7	Frantiska	Brno	MendelU	19/02/1996
7	1	Otto	Sobkovice	-	18/03/2000
8	8	Ondrej	Brno	VUT	11/12/1999
9	9	Patrik	Brno	VUT	11/02/1998
10	10	Adam	Brno	HON	01/11/1996
11	14	Pavel	Brno	eON	02/02/1999



## E Ukázky testovacích skriptů

### E.1 Hlavní testovací skript

```
30 test_result_t test_station_main_test_script(void){
31     if (!_read_pcb_config()) {
32         test_result.return_code = ERROR_NOT_INITIALIZED;
33     }
34     else {
35         if (!_test_serial_com()) {
36             _measure_selected_adc_channels();
37             _generate_final_report();
38         }
39         else {
40             _activate_digital_outputs();
41             _measure_all_adc_channels();
42             _activate_digital_inputs();
43             _check_digital_input_states();
44             _generate_final_report();
45         }
46     }
47     return test_result;
48 }
```

### E.2 Načtení názvu DPS z její konfigurace v EEPROM

```
50 bool _read_pcb_config(void){
51     at24c_activate();
52     test_result.return_code =
53         at24c_sequential_read(PCB_CFG_DPS_NAME_ADDR, (uint8_t *)
54             pcb_cfg_eeprom.dps_name, PCB_CFG_DPS_NAME_SIZE);
55     .
56     .
57     .
58     if (test_result.return_code == ERROR_NONE){ return true; }
59     else{ return false; }
60 }
```

## E.3 Změření všech kanálů ADC

```
97 void _measure_all_adc_channels(void){
98     uint8_t adc_buffer_raw[ADS7830_N_CHNL];
99
100     ads7830_activate();
101     test_result.return_code =
        ads7830_measure_all_channels_SE(PDIROFF_ADCON,
        adc_buffer_raw);
102     if (test_result.return_code == ERROR_NONE){
103         for (uint8_t chnl = 0; chnl < ADS7830_N_CHNL; ++chnl){
104             test_station.adc_buffer_volt[chnl] =
                _adc_raw_to_voltage(adc_buffer_raw[chnl], chnl);
105             test_result.adc_channel_passed[chnl] =
                _verify_test_criteria_voltage(TEST_CRIT_VOLT_RANGE_PERC,
106             chnl, test_station.adc_buffer_volt[chnl]);
107         }
108     }
109     else { io_write(edbg_io, (uint8_t *)"Voltage measurement
        error!\n", 27); }
110 }
```

## E.4 Aktivace dig. vstupů pro testovanou DPS

```
194 void _activate_digital_inputs(void){
195     /* Activate digital inputs via GPIO expander */
196     // Set GPIO Expander Slave Address
197     pca9557_activate();
198
199     // Set IO0 and IO1 as inputs, others as outputs
200     pca9557_set_mode_all(IO_OUTPUT);
201     pca9557_set_pin_mode(IO0, IO_INPUT);
202     pca9557_set_pin_mode(IO1, IO_INPUT);
203
204     // Set all pins to High (no effect on pins defined as inputs)
205     pca9557_set_state_all(IO_HIGH);
206 }
```