



# 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

## SYSTÉM PRO MĚŘENÍ A ŘÍZENÍ TLAKU VE VAKUOVÉ APARATUŘE

SYSTEM FOR MEASUREMENT AND CONTROL OF PRESSURE IN VACUUM APPARATUS

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Jan Sedlář

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Karel Juřík

BRNO 2023

# Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Jan Sedlář

**ID:** 228521

**Ročník:** 3

**Akademický rok:** 2022/23

**NÁZEV TÉMATU:**

## **Systém pro měření a řízení tlaku ve vakuové aparatuře**

**POKyny PRO VYPRACOVÁNÍ:**

1. Seznamte se s principy fungování vakuových měrek, vakuových vývěv a ventilů. Zpracujte stručnou rešerši.
2. Seznamte se s fungováním A/D převodníků, vyberte vhodný převodník pro používanou vakuovou měрку.
3. Navrhněte systém řízení tlaku v aparatuře.
4. Vyberte vhodnou řídicí jednotku a navrhněte uživatelské rozhraní.
5. Návrh zrealizujte a otestujte, diskutujte jeho funkcionalitu.
6. Výsledky zpracujte do bakalářské práce.

Na semestrální práci je nutné vypracovat body č. 1 až 4. zadání.

**DOPORUČENÁ LITERATURA:**

HNATEK, Eugen R., 1976. A user's handbook of D/A and A/D converters. Wiley. ISBN 978-0471401094.

**Termín zadání:** 6.2.2023

**Termín odevzdání:** 22.5.2023

**Vedoucí práce:** Ing. Karel Juřík

**doc. Ing. Václav Jirsík, CSc.**  
předseda rady studijního programu

**UPOZORNĚNÍ:**

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

## **ABSTRAKT**

Cílem bakalářské práce je pro dostupné vakuové komponenty a aparaturu vytvořit systém pro měření a řízení tlaku plynu v této aparatuře. Po rešerších v úvodních částech práce, které byly zpracovány pro lepší pochopení problematiky a stanovení vhodných řešení problému, byl na základě teoretického návrhu řešení cíl práce realizován. V kapitolách práce je rozebírána volba vhodných komponent a jejich následné začlenění do funkčního celku. Jsou popisovány hardwarová i softwarová řešení, která byla v práci realizována. Automatizované řízení tlaku v aparatuře je vystavěno na mnohých podpůrných zdrojích hardwarového i softwarového charakteru jako zařízením automaticky vytvořený regulátor, který obsluhuje umožňuje snadné dosažení požadovaného pracovního bodu pro realizaci dalšího výzkumu ve vakuové aparatuře.

## **KLÍČOVÁ SLOVA**

řízení tlaku, vakuová aparatura, automatická identifikace soustavy, automatizovaný návrh regulátoru, řízení vakuových periférií

## **ABSTRACT**

The aim of the bachelor's thesis is to create a system for measurement and control of gas pressure with available vacuum components in vacuum apparatus. After conducting research in the introductory sections of the thesis to better understand the issue and determine suitable problem-solving approaches, the goal of the thesis was implemented based on the theoretical design. The chapters of the thesis discuss the selection of appropriate components and their integration into the functional unit. Both hardware and software solutions implemented in the thesis are described. The automated pressure control in the apparatus is built upon multiple supporting resources of hardware and software nature, which combined create an automatically designed controller that enables easy achievement of the desired operating point for further research in the vacuum apparatus.

## **KEYWORDS**

pressure control, vacuum apparatus, automatic system identification, automated controller design, control of vacuum peripherals

SEDLÁŘ, Jan. *Systém pro měření a řízení tlaku ve vakuové aparatuře*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2023, 93 s. Bakalářská práce. Vedoucí práce: Ing. Karel Juřík,

## Prohlášení autora o původnosti díla

**Jméno a příjmení autora:** Jan Sedlář

**VUT ID autora:** 228521

**Typ práce:** Bakalářská práce

**Akademický rok:** 2022/23

**Téma závěrečné práce:** Systém pro měření a řízení tlaku ve vakuové aparatuře

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

Děkuji vedoucímu mé bakalářské práce panu Ing. Karlu Juříkovi za odborné vedení, konzultace, vstřícnost a podnětné návrhy k práci.

Dále také panu doc. Ing. Miloslavu Steinbauerovi, Ph.D. za odborné konzultace

Děkuji společnosti SpaceLabEU, SE, za jejíž podpory byla práce vytvořena.

# Obsah

Úvod	13
<b>1 Vakuové komponenty</b>	<b>14</b>
1.1 Měrky	14
1.2 Vývěvy	17
1.3 Ventily	23
<b>2 Analogově-digitální převodníky</b>	<b>24</b>
2.1 Flash převodníky	25
2.2 Half-flash převodníky	26
2.3 Převodníky s postupnou aproximací	26
2.4 Převodníky s rampou	27
2.5 Převodník $\Sigma\Delta$	27
<b>3 Návrh řízení systému</b>	<b>31</b>
<b>4 Řídící jednotka</b>	<b>34</b>
<b>5 Popis systému</b>	<b>37</b>
5.1 Uživatelské rozhraní	39
<b>6 Hardwarové zdroje</b>	<b>43</b>
6.1 Desky tištěných spojů a jejich propojení	43
6.2 Napájení	44
6.3 Komunikace s turbomolekulární vývěvou	46
6.4 Komunikace s vakuovým ventilem	47
6.5 Vyčítání hodnot tlaku a identifikace měrek	50
6.6 Slot SD karty	51
6.7 Modul reálného času	52
6.8 Konektory a fyzická propojení	53
6.9 Napájení a propojení s displejem	54
6.10 Kryt zařízení	54
<b>7 Softwarové zdroje</b>	<b>56</b>
7.1 Pfeiffer Vacuum Protocol	56
7.2 Ovládání vakuového ventilu	59
7.3 Obsluha vakuových měrek	63
7.3.1 Identifikace připojených měrek	63
7.3.2 Stanovení hodnot měřeného tlaku	64

7.4	Ukládání dat na SD kartu . . . . .	65
7.5	Reálný čas . . . . .	66
7.6	Komunikace s displejem . . . . .	66
<b>8</b>	<b>Činnosti systému</b>	<b>70</b>
8.1	S krátkou periodou . . . . .	71
8.1.1	Obsluha displeje . . . . .	71
8.2	S dlouhou periodou . . . . .	72
8.2.1	Vyčítání hodnot tlaku . . . . .	72
8.2.2	Zpracování požadavku na cílovou hodnotu tlaku v aparatuře .	73
8.2.3	Povelování vakuových komponent . . . . .	73
8.3	Činnosti systému volané aperiodicky . . . . .	73
8.3.1	Zapnutí a vypnutí zařízení . . . . .	73
8.3.2	Identifikace vakuového systému . . . . .	74
8.3.3	Návrh regulátoru . . . . .	79
<b>9</b>	<b>Validace automatické identifikace a regulace</b>	<b>85</b>
	<b>Závěr</b>	<b>89</b>
	<b>Literatura</b>	<b>90</b>



# Seznam obrázků

1.1	Diagram měření tlaku pomocí výchylky membrány [12] . . . . .	15
1.2	Schéma vnitřního zapojení Piraniho sondy s konstantním příkonem. $R_W$ : odpor měřicího drátu [12] . . . . .	16
1.3	Diagram orientace elektrického pole ve vakuové měrce. $B$ : kolmé magnetické pole, šedá oblast: oblast prostorového náboje [12] . . . . .	17
1.4	Příklady rotačních vakuových vývěv s jednou osou rotace [13] . . . . .	18
1.5	Příklady rotačních vakuových vývěv se dvěma osami rotace [13] . . . . .	19
1.6	Diagram proudové vakuové vývěvy [14] . . . . .	20
1.7	Řez turbomolekulární vývěvou se spodní částí Holweckova typu [15] . . . . .	21
1.8	Rotor a stator turbomolekulární vývěvy Holweckova typu [14] . . . . .	21
1.9	Kryopumpa [16] . . . . .	22
1.10	Diagram úhlového ventilu [17] . . . . .	23
2.1	3bitový flash převodník. $U_{S/H}$ : navzorkované napětí, $U_{ref}$ : referenční napětí [9] . . . . .	25
2.2	Blokový diagram 8bitového half-flash převodníku [9] . . . . .	26
2.3	Princip stanovení výsledku převodu z časového intervalu u převodníků s rampou [9] . . . . .	28
2.4	Základní blokové schéma sigma-delta převodníku [9] . . . . .	29
2.5	Znázornění typických chyb převodu [9] . . . . .	29
3.1	Blokové schéma systému pro řízení tlaku ve vakuové aparatuře . . . . .	33
5.1	Vacuum System Controller připojen k vakuové aparatuře . . . . .	37
5.2	Logická struktura popisu VSC . . . . .	38
5.3	Uživatelské rozhraní – Hlavní stránka . . . . .	39
5.4	Uživatelské rozhraní – Klávesnice pro nastavení požadované hodnoty tlaku v aparatuře . . . . .	40
5.5	Uživatelské rozhraní – Karta vakuových periferií (vývěvy a ventilu) . . . . .	40
5.6	Uživatelské rozhraní – Manuální volba připojené vakuové měrky . . . . .	41
5.7	Uživatelské rozhraní – Karta uživatelských preferencí . . . . .	42
5.8	Uživatelské rozhraní – Nastavení řízení vakuové aparatury . . . . .	42
6.1	Deska tištěných spojů s doplňkovými obvody . . . . .	43
6.2	Konektorová deska tištěných spojů . . . . .	44
6.3	Konektor KPJX-4S z produktového listu vhodného napájecího zdroje GSM220A [21] . . . . .	44
6.4	Schéma zapojení vstupní ochrany a 5V DC-DC měniče . . . . .	45
6.5	Schéma zapojení 3,3V DC-DC měniče . . . . .	45
6.6	Schéma zapojení obvodu pro komunikaci s turbomolekulární vývěvou . . . . .	46
6.7	Požadavky na komunikaci s vakuovým ventilem [2] . . . . .	47

6.8	Schéma zapojení obvodu pro komunikaci s vakuovým ventilem . . . .	48
6.9	Schéma zapojení obvodu pro analogové ovládání vakuového ventilu . .	48
6.10	Výňatek z produktového listu OZ LM741 [35] . . . . .	49
6.11	Výňatek z produktového listu OZ OPA189 [36] . . . . .	49
6.12	Schéma zapojení analogově-digitálního převodníku . . . . .	50
6.13	Schéma zapojení modulu pro připojení SD karty . . . . .	52
6.14	Schéma zapojení modulu reálného času . . . . .	52
6.15	Umístění konektorů na konektorové desce . . . . .	54
6.16	Zkompleťovaný systém v krytu VSC . . . . .	55
7.1	Formát zpráv <i>Pfeiffer Vacuum Protocolu</i> [27] . . . . .	56
7.2	Produktový list popisující zprávy pro ovládání ventilu [2] . . . . .	60
7.3	Vyžádání informací od ventilu [2] . . . . .	60
7.4	Vztah pro přepočítání skutečné hodnoty tlaku podle plynu přítomného v aparatuře [1] . . . . .	64
7.5	Převodní vztahy $U \rightarrow p$ pro vakuové měřky . . . . .	65
7.6	Příklad zaslání specifických znaků pro spuštění <code>trigger</code> funkcí po stisku tlačítka na displeji . . . . .	69
8.1	Rozdělení činností programů podle jejich periodicity . . . . .	70
8.2	Parametry systému určované z přechodové charakteristiky . . . . .	75
8.3	Blokové schéma stavového automatu identifikace vakuové aparatury .	76
8.4	Blokové schéma implementovaného diskrétního regulátoru [30] . . . .	81
8.5	Srovnání metod filtrací derivační složky [30] . . . . .	82
9.1	Identifikace reálného systému . . . . .	86
9.2	Akční zásah regulátoru . . . . .	87
9.3	Vývoj složek S, P a D regulátoru . . . . .	87

# Seznam tabulek

4.1	Shrnutí vlastností desek Arduino [6] . . . . .	34
6.1	Hodnoty identifikačních odporů testovaných měrek [28][1] . . . . .	51
8.1	Postup pro návrh regulátoru pomocí metody Ziegler-Nichols [10] . . .	74
8.2	Postup pro návrh regulátoru pomocí metody Ziegler-Nichols při sta- novení kritických parametrů z přechodové charakteristiky [11] . . . .	75
8.3	Souhrnný přehled parametrů použitého $\beta$ -PSD regulátoru . . . . .	83
9.1	Naměřená identifikační data pro počáteční tlak $p_0 = 19,432$ mPa a akční zásah 750→1500 Hz . . . . .	85
9.2	Vstupy, stavy a výstupy regulátoru při $w = 12$ mPa, $T = 3$ s . . . . .	88

## Seznam výpisů

7.1	PVP: metoda <code>controlCommand</code> . . . . .	57
7.2	PVP: <code>readResponse</code> . . . . .	58
7.3	ValveProtocol: Metody pro čtení a zápis jednoho bytu . . . . .	61
7.4	ValveProtocol: Metody pro odesílání a přijímání zpráv . . . . .	61
7.5	ValveProtocol: nastavení ventilu do konkrétní pozice . . . . .	62
7.6	gauge: Identifikace připojených měrek . . . . .	63
7.7	gauge: stanovení tlaku z napěťové hodnoty . . . . .	65
7.8	Tvorba textového řetězce aktuálního času . . . . .	66
7.9	Kontrola připojených periférií . . . . .	67
7.10	Trigger funkce volaná při změně připojené pumpy nebo ventilu . . . . .	69
8.1	Funkce volaná, pokud je na displeji otevřena úvodní stránka . . . . .	72
8.2	Identifikace systému: stav <code>prepare</code> . . . . .	77
8.3	Identifikace systému: stav <code>step</code> . . . . .	77
8.4	Identifikace systému: stav <code>systemIdent</code> . . . . .	78
8.5	Identifikace systému: stav <code>regConstruct</code> . . . . .	79
8.6	Regulátor: periodický výpočet akčního zásahu . . . . .	84

# Úvod

Cílem bakalářské práce je pro dostupné vakuové komponenty a aparaturu vytvořit systém pro měření a řízení tlaku plynu v této aparatuře. Po rešerších v úvodních částech práce, které byly zpracovány pro lepší pochopení problematiky a stanovení vhodných řešení problému, byl na základě teoretického návrhu řešení cíl práce realizován. V kapitolách práce je rozebírána volba vhodných komponent a jejich následné začlenění do funkčního celku. Jsou popisovány hardwarová i softwarová řešení, která byla v práci realizována. Automatizované řízení tlaku v aparatuře je vystavěno na mnohých podpůrných zdrojích hardwarového i softwarového charakteru jako zařízením automaticky vytvořený regulátor, který obsluze umožňuje snadné dosažení požadovaného pracovního bodu pro realizaci dalšího výzkumu ve vakuové aparatuře.

# 1 Vakuové komponenty

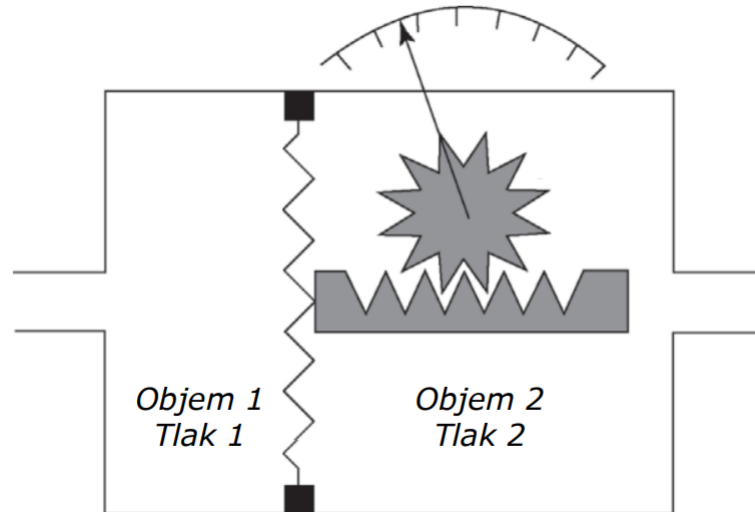
## 1.1 Měrky

Měření tlaku se ve vakuových aparaturách provádí dvěma způsoby – buď jako měření přímé zejména pomocí stanovení velikosti síly působící na plochu o známých rozměrech, jak ukazuje Rce. (1.1), anebo měření nepřímé, kam spadají například měření hustoty částic nebo tepelné vodivosti.

$$p = \frac{F}{A} \quad (1.1)$$

Nejdříve uvedu měrky určené pro přímá měření tlaku, které se dají rozdělit do dvou základních kategorií podle použitého měřicího mechanismu. Zaprvé se jedná o měrky mechanické využívající již zmíněného vztahu mezi silou působící na plochu a tlakem. Standardní provedení obsahuje komoru rozdělenou na dvě části pomocí membrány. V jedné z těchto částí se nachází měřený plyn a v druhé bychom našli jiný plyn s referenčním tlakem. Na membránu o známé velikosti je potom vyvíjena síla úměrná rozdílu tlaků obou plynů. Touto silou je způsobena výchylka membrány, kterou můžeme měřit a stanovit z ní tak výsledný tlak. Tento princip je znázorněn na Obr. 1.1. V závislosti na zvolené referenci se vlastně jedná o rozdílové měření tlaků mezi použitými dvěma plyny, v praxi se však dá využít jako reference plynu se zanedbatelně nízkým tlakem. Ten tak lze v rámci měření považovat za nulový, protože je menší, než je rozlišovací schopnost měřidla. Rozdělení těchto měrek provádíme právě podle zvoleného typu reference na ty s referencí v podobě atmosférického tlaku se senzorem výchylky na straně reference, s referencí v podobě nulového tlaku se senzorem na straně měření nebo také s nulovou referencí ale se senzorem na straně reference. Měrky tohoto typu se často nabízejí v provedení s fyzickou ručkou a ciferníkem, na který se přímo přenáší výchylka membrány, anebo jako měřidlo určené k podmíněnému spouštění ostatních komponent systému, jako jsou pumpy, ventily nebo topení. Podmínkou je potom dosažení či překročení stanoveného tlaku.

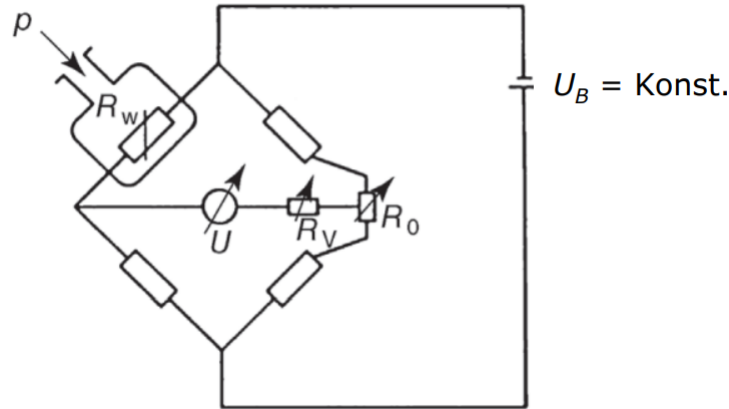
Druhým typem měrek sloužících k přímým měřením jsou měrky využívající efektu plynového tření. Typické provedení potom sestává z nerezové koule uvnitř trubice, která je připojena k aparatuře, a tak naplněna plynem o měřeném objemu. Koule je v trubici uchycena bezkontaktně pomocí několika magnetů, aby se zajistila eliminace parazitních mechanických tření majících vliv na výsledky měření, a dalšími magnety je potom tato koule roztáčena. Mezi rotující koulí a statickým plynem pak vzniká třecí síla, která je úměrná tlaku tohoto plynu. Touto metodou je obtížné měřit vyšší tlaky ( $> 1 \text{ Pa}$ ), protože zde začínají mít výrazný vliv mezičásticové srážky plynu.



Obr. 1.1: Diagram měření tlaku pomocí výchylky membrány [12]

Měrky pro nepřímá měření tlaku plynů je opět možné rozdělit na dvě základní kategorie: ionizační a využívající tepelné vodivosti. V měrkách využívajících prvního zmíněného jevu dochází k měření tepelných ztrát nějakého zahřátého objektu, nejčastěji potom rozžhaveného drátu. Ten bývá součástí *wheatstoneova mostu*, který zajišťuje jak jeho napájení, tak i měření změny jeho odporu nebo rozptýleného tepelného výkonu. Tento typ měrky se často nazývá po svém vynálezci Marcellu Piranim jako Piraniho sonda a její schématické znázornění je zobrazeno na Obr. 1.2. K samotnému měření potom existuje hned několik přístupů. Drát je možné udržovat při konstantní teplotě pomocí změn příkonu, ze kterého lze stanovit tlak plynu. Výhodou je vysoká linearita na širokém rozsahu tlaků i velmi přesné měření, ale jedná se o značně komplexní uspořádání s vysokou pořizovací cenou. Alternativou je potom naopak konstantní příkon a sledování změn jiných veličin jako teploty zahřívajícího drátu nebo kompenzačního proudu mostem. Tyto přístupy nabízejí menší rozsah měření, ale jsou jednodušší na výrobu, a proto také levnější. Hlavní uplatnění tyto sondy naleznou v oblasti *jemného vakua* a obecně vynikají svou rychlou odezvou a ve většině provedení slouží pro hrubější měření tlaku. Potom jsou Piraniho sondy také obecně levné a díky tomu i velmi rozšířené.

V případě ionizačních měření se využívá úměry mezi hustotou částic plynu (a tedy tlaku) a některou z elektrických veličin. Za těmito účely je plyn v hlavě měrky částečně ionizován a posléze dochází buď k měření iontového proudu (v měrce s termokatodou) nebo k měření proudu výboje (v měrce s kříženými elektromagnetickými poli). V případě měrek s termokatodou (z angl. hot cathode) jsou elektrony emitovány z rozžhavené katody a vnějším polem urychlovány k anodě, která ji obklopuje. Elektrony se na své cestě mezi dvěma elektrodami srážejí s částicemi plynu



Obr. 1.2: Schéma vnitřního zapojení Piraniho sondy s konstantním příkonem.  $R_w$ : odpor měřicího drátu [12]

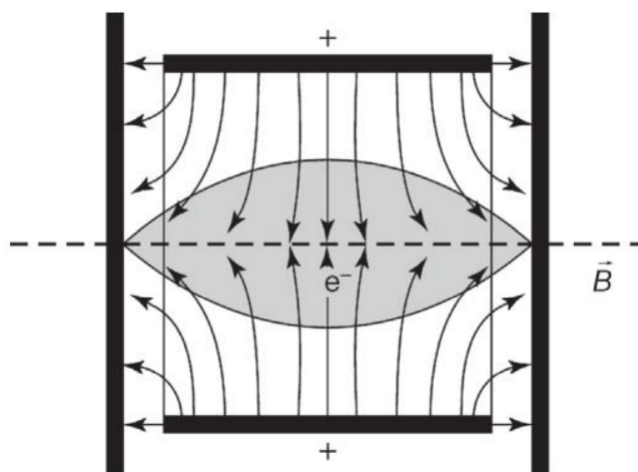
a s určitou pravděpodobností je mohou ionizovat. Na sběrači potom dochází k zachycování těchto iontů a k měření iontového proudu, který je úměrný hustotě částic plynu. Pro měřky podobného typu se také často užívá termínu *cold cathode*. Tyto měřky pro emitování elektronů využívají tzv. *cold field* difúze, funkčností jsou ale termokatodám velmi podobné.

U měrek s kříženými elektromagnetickými poli se snažíme vytvořit stejnosměrný výboj mezi dvěma elektrodami s odlišným elektrickým potenciálem (napětí v řádu několika kilovoltů). Druhé (křížené) pole vzniká kolmo na spojnici těchto dvou elektrod, tak jak ukazuje Obr. 1.3, a má charakter magnetického pole. Jeho účel je prodloužit dráhu elektronů putujících od jedné z elektrod ke druhé a zvýšit tak pravděpodobnost jejich srážky s částicemi plynu. Měřený výbojový proud je potom závislý na tlaku plynu. Důležité je si však uvědomit, že v tomto případě je měřený tlak související s použitým plynem. Ionizační měřky s emitující katodou se volí zejména v případech, kdy je třeba provádět přesná měření (chyba měření je menší než deset procent), a dají se tak využít jako reference. Naopak měřky s kříženými poli jsou poměrně náchylné na rušení vnějšími faktory a uplatnění naleznou především díky své nižší ceně.

V praxi není možné zvolit jednu měřku pro přesné a rychlé měření, které je možné provádět na celém spektru měřitelných tlaků. Častým řešením je potom integrace více dílčích měrek do jednoho těla společně se systémem, který mezi nimi dle aktuálního tlaku přepíná. Vyhlazením výstupního signálu při přepínání mezi jednotlivými měrkami se potom zabývají speciální elektronické obvody zabudované v takovýchto měrkách.

Všechny měřky zmíněné v této kapitole slouží k měření celkového tlaku ve vakuové aparatuře. Existuje však i celá řada sond, které jsou určeny k měření tlaků





Obr. 1.3: Diagram orientace elektrického pole ve vakuové měrce.  $B$ : kolmé magnetické pole, šedá oblast: oblast prostorového náboje [12]

dílčích plynů v takové aparatuře obsažených. Takové měření ale není pro projekt tohoto typu v aktuální chvíli důležité, a proto se jimi nebudu dále zabývat. [12]

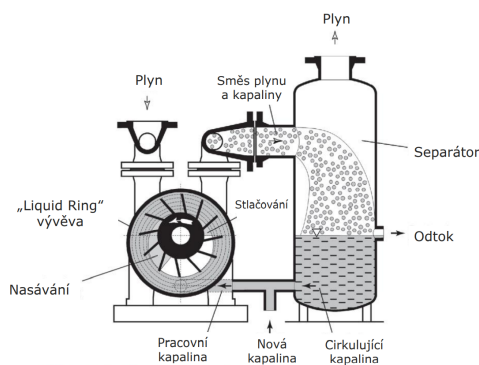
## 1.2 Vývěvy

Velmi rozšířeným typem vakuové vývěvy jsou objemová čerpadla. Jedná se o pumpy pracující v opakovaných cyklech, při kterých dochází k postupnému odčerpávání plynu z aparatury za pomoci pístů, rotorů a podobně. Cyklus se skládá ze dvou částí, kdy v první dochází k nasátí plynu do vývěvy a v druhé části naopak dochází k vyfouknutí plynu výfukovým ventilem. Jedná se o vývěvy vhodné pro hrubé a jemné vakuum a lze je rozdělit na oscilační a rotační. U oscilačních dochází, jak již jejich název napovídá, k nasávání plynu do vývěvy pomocí nějakého oscilujícího tělesa. Většinou se jedná o píst nebo membránu. Plyn může a nemusí při svém průchodu takovouto pumpou přijít do styku s oleji a mazivou, které umožňují plynulý chod zařízení. V případě, že ke kontaminaci plynu těmito složkami nedochází, pumpu nazýváme jako suchoběžnou (*dry pump*). U rotačních vývěv rozlišujeme podél kolika os dochází k rotaci. Pro každou z kategorií uvedu několik příkladů konkrétního řešení, jakým dochází k čerpání plynu. Do kategorie vývěv s jednou osou rotace, z nichž vybrané příklady uvádím na Obr. 1.4, bychom zařadili tyto:

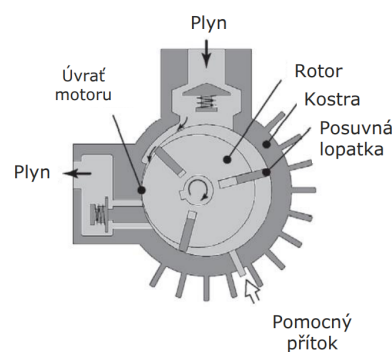
- *Liquid ring* – předávání hybnosti plynu pomocí kapaliny rotující po obvodu válce.
- Vývěvy s posuvnými lopatkami – rotor je v takovéto pumpě umístěn uvnitř statoru, ale nevystředěně. Prostor statoru je potom výsuvnými lopatkami rotoru dotýkajícími se vždy stěn rozdělen na několik nestejně velkých objemů.

Při rotaci se tyto vytyčené prostory objemově mění a dochází tak k nasávání plynu do nich a následného vytlačování plynu z nich ven.

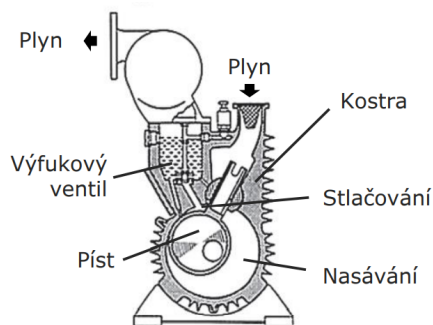
- Vývěvy s rotujícím pístem – jedná se o pumpy na podobném principu jako při posuvných lopatkách. Prostor je však rozdělen pomocí speciálně tvarovaného válce.
- Trochoidní vývěvy – změny objemu v rámci statoru jsou u těchto pump zajišťovány eliptickým pístem.
- Spirálové kompresory – takováto pumpa je tvořena dvěma koaxiálně umístěnými Archimédovými šrouby zapuštěnými jeden v druhém. První z nich je statický a druhý se v rámci vřele mezi závitů šroubů otáčí. Tímto pohybem tak dochází k pumpování plynu.



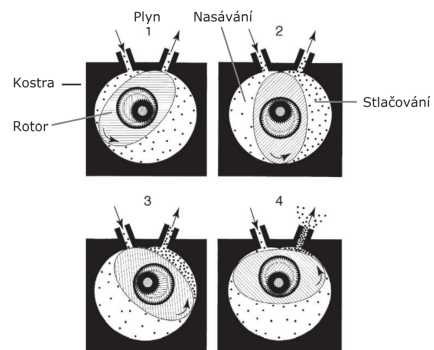
(a) „Liquid Ring“ vývěva



(b) Vývěva s posuvnými lopatkami



(c) Vývěva s rotujícím pístem



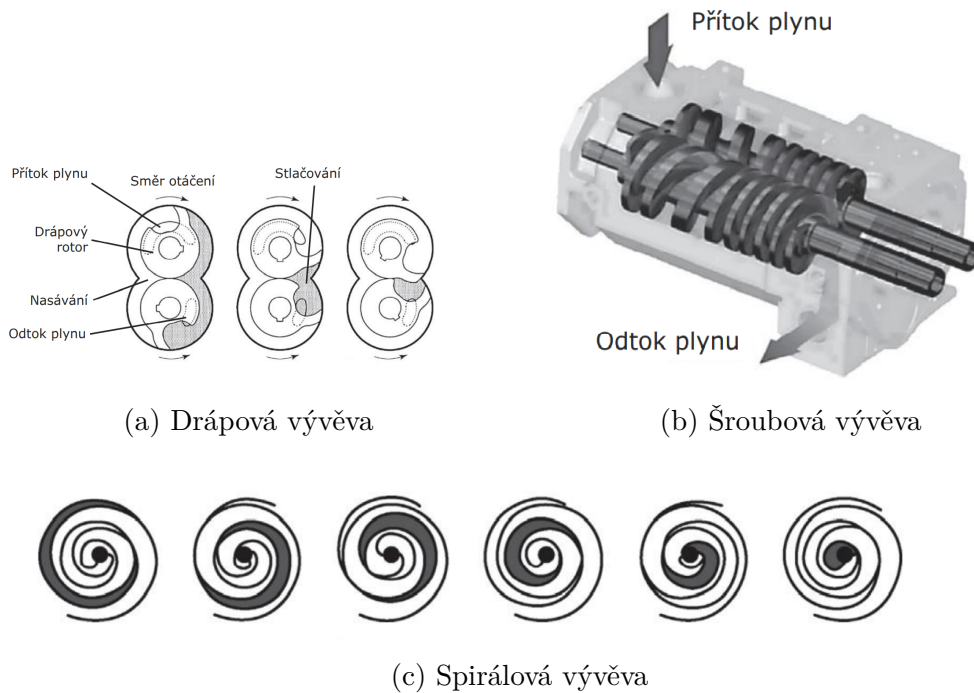
(d) Trochoidní vývěva

Obr. 1.4: Příklady rotačních vakuových vývěv s jednou osou rotace [13]

Podobně můžeme shrnout rotační pumpy se dvěma osami, kde rozlišujeme tři významné kategorie, jejichž nákresy jsou zobrazeny na Obr. 1.5.

- Šroubové – sestávají se ze dvou rotorů připomínajících šrouby, které jsou umístěny v těsné blízkosti. Každý z rotorů má jiný směr otáčení, čímž po své délce

- vytvářejí několik komor, které se pohybují od jednoho konce rotorů na druhou.
- Dráповé čerpadlo – Na dvou osách jsou umístěny rotory, které svým tvarem připomínají zahnutý zvířecí dráp. Opět dochází k jejich otáčení v opačných směrech, čímž dochází v prostoru dvojitého statoru ke vzniku komor o proměnlivých objemech. Těch se využívá k čerpání plynu.
  - Rootsova vývěva – uvnitř oválné šachty dochází k otáčení dvou rotorů, které svým tvarem připomínají fazoli anebo číslici osm. Samotný princip čerpání plynu je potom totožný s předchozím případem.



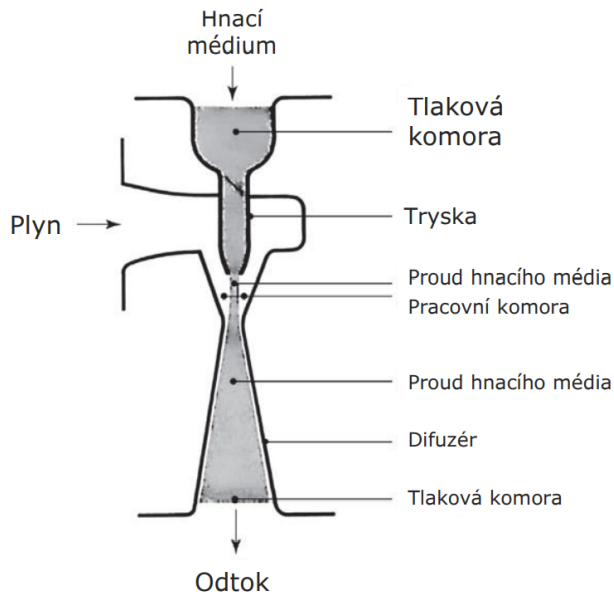
Obr. 1.5: Příklady rotačních vakuových vývěv se dvěma osami rotace[13]

Obecně mají objemová čerpadla velmi široké uplatnění, což má za následek jejich všeobecnou rozšířenost. Nevýhodou je jejich nevhodnost použití s plyny, které kondenzují při teplotě přibližně 60–90 °C, což jsou obvyklé pracovní teploty těchto pump. [13]

V případě *proudových pump* je plyn čerpán pomocí proudu hnací tekutiny, která pumpou prochází. Podle zvoleného pracovního média rozlišujeme tři základní typy proudových pump: využívající kapalinu, plyn anebo výpary. Proud média vzniká vypuštěním hnací tekutiny z tlakové nádoby do pracovní komory, kde dochází k přimíchání čerpaného plynu a předání hybnosti proudu částicím tohoto plynu. Celý proces probíhá podle známé Bernoulliho rovnice Rce. (1.2).

$$\frac{1}{2}\rho v^2 + p + \rho gh = \text{konst.} \quad (1.2)$$

Proudové pumpy můžeme rozdělit na dvě podkategorie podle pracovního tlaku. Jedná se buď přímo o proudové, kde tlak média v pracovní komoře je srovnatelný s tlakem v této komoře a k předávání hybnosti dochází pomocí jakési nerovné turbulentní hranice hnacího média a čerpaného plynu. Nebo se jedná o pumpy



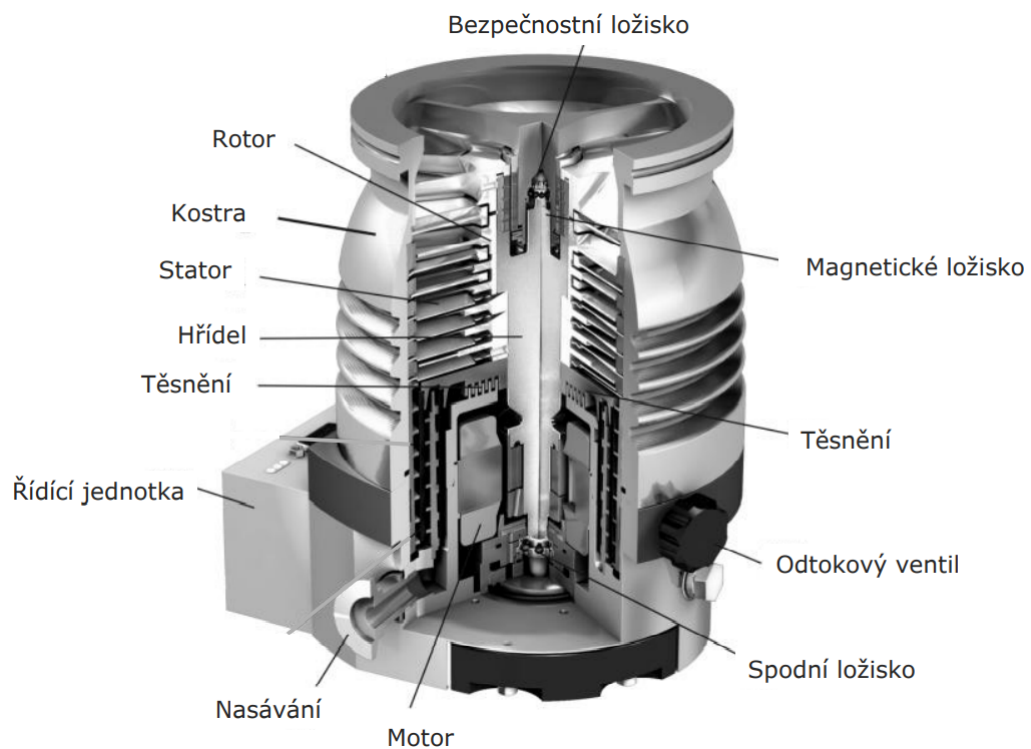
Obr. 1.6: Diagram proudové vakuové vývěvy [14]

práce s materiály nebo úprava povrchů, elektronová mikroskopie, vakuové zámky, výroba polovodičů či urychlovače částic a jaderná fúze. Potřebnou hybnost částicím plynu dodávají velmi rychle se pohybující rotory. Pumpy turbomolekulárního typu obvykle nejsou schopny vyfukovat čerpaný plyn přímo do atmosféry a potřebují tak ještě další pumpu (podpůrné čerpadlo), která plyn stlačí na potřebný tlak. K těmto účelům se nejčastěji využívají objemová čerpadla. Na rozdíl od například sorpčních pump, které budou rozebírány vzápětí, nevyžadují pracovní přestávky a jsou schopné začít čerpat velmi rychle. Konkrétní provedení se obvykle pojmenovává podle jejich vynálezců jako Beckerova, Krugerova nebo Holweckova. Často lze také nalézt vývěvy kombinující několik typů v jednom těle, jako je vidět na Obr. 1.7, kde lze ve spodní části zařízení pozorovat typické *Holweckovo uspořádání*. [15] Detailní zobrazení statoru takovéto složené turbomolekulární vývěvy a statorů Holweckova typu lze zase vidět na Obr. 1.8.

Sorpční pumpy využívají fyzikálního jevu *sorpce* a plyn tak čerpají pomocí navázání jeho molekul k nějakému povrchu. Nedochozí tak vlastně k samotnému čerpání ale pouze jakémusi uvěznění plynu. Své uplatnění tyto pumpy nalézají při všech

difúzní, kdy tlak pracovního média je výrazně vyšší než tlak v pracovní komoře a k míchání plynu a média dochází vlivem difúze. Zejména vývěvy využívající páru jako hnacího média mají velmi širokou aplikaci a jsou hojně používané, často také ve vícestupňových zapojení. [14] Jednoduché znázornění proudové pumpy je zobrazeno na Obr. 1.6.

*Turbomolekulární vývěvy* nalézají velmi široké uplatnění v aplikacích pracujících zejména s vysokým vakuem. Konkrétně se může jednat o aplikace jako



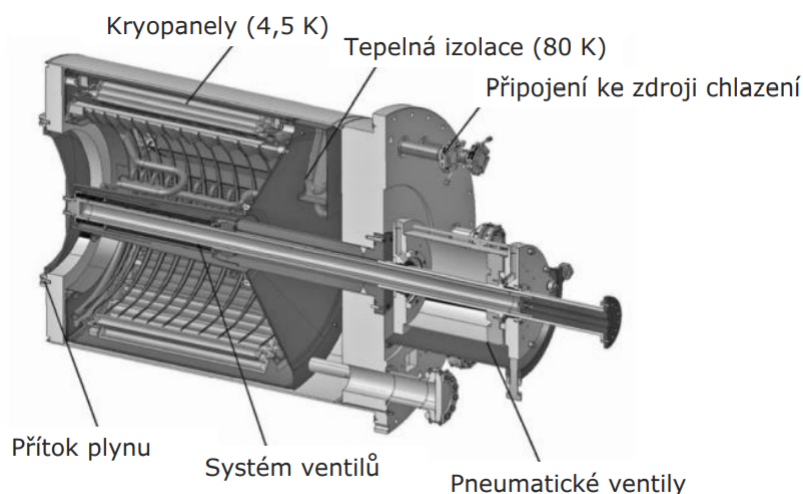
Obr. 1.7: Řez turbomolekulární vývěvou se spodní částí Holweckova typu [15]



Obr. 1.8: Rotor a stator turbomolekulární vývěvy Holweckova typu [14]

tlacích využívaných ve vakuových aparaturách, ale především se používají v oblasti ultra vysokého vakua. Podle konkrétní realizace se tyto vývěvy dají rozdělit na adsorpční a getrové. Adsorpční využívají vlastnosti některých pevných látek, které jsou za nižších teplot schopné na svůj povrch vázat relativně velká množství plynu. Getrové naproti tomu využívají speciálních materiálů – tzv. getrů – které chemicky naváží částice plynu. Ty mají obecně poměrně vysoké rychlosti „čerpání“ i za nízkých tlaků a jejich největší výhodou je, že při své činnosti neprodukuje vibrace. Uplatnění nacházejí ve stejných odvětvích jako vývěvy turbomolekulární. U getrových pump dále rozlišujeme, zda ke své činnosti využívají anebo nevyužívají ionizaci. Neionizační getrové vývěvy se někdy provozují s tzv. *vypařovaným getrem*, kde dochází k neustálému obnovování povrchu, který slouží k vázání částic plynu. Tyto pumpy však nejsou schopny pracovat se vzácnými plyny nebo třeba s metanem, který je obecně méně chemicky reaktivní. Zejména z tohoto důvodu tak posléze došlo k vývoji getrových pump využívající i ionizaci plynu. Ionizované molekuly je totiž možné urychlovat vnějším polem a povrch getru jimi ostřelovat, čímž snáze dojde k navázání těchto molekul k povrchu. [18]

Poslední skupinou rozšířených vakuových vývěv jsou *kryopumpy*, které využívají nízkých teplot ( $< 120$  K). K vytvoření vakua v aparatuře dojde změnou skupenství plynu na kapalné nebo pevné poté, co je plynu odebráno dostatečné množství tepla pomocí instalovaných kryopanelů, které lze vidět na Obr. 1.9 zobrazující řez kryopumpou. Toho se obvykle dosahuje cyklickým odběrem tepla pomocí pracovního média, kterým často bývá hélium. Pumpy tohoto typu se často využívají v soustavách s vakuovou tepelnou izolací, jako je skladování zkapalněných plynů nebo pro supravodivé materiály. [16]



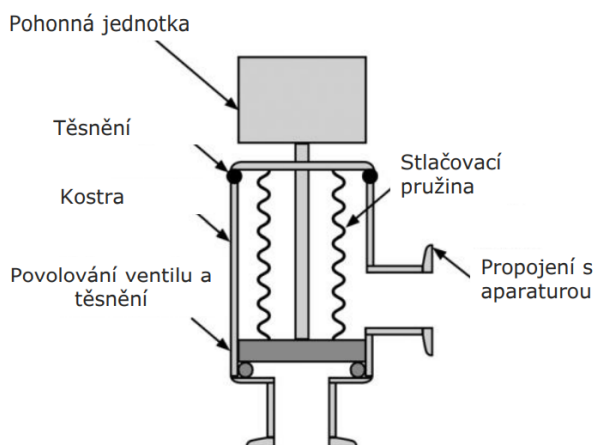
Obr. 1.9: Kryopumpa [16]

## 1.3 Ventily

Pro spojování dvou systémů s plyny o rozdílných tlacích se využívají vakuové ventily. Mohou nabývat buď pouze dvou stavů (otevřený a zavřený) nebo mohou operovat v libovolných pozicích mezi těmito dvěma stavy. Mezi základní vlastnosti ventilů patří průtok, úniky plynu, životnost, použité materiály nebo velikost ventilu. Obecně rozlišujeme celou řadu typů ventilů, ale teorie k nim není nikterak obsáhlá, a proto jejich základní vlastnosti uvedu výčtovým způsobem:

- Úhlový 90 – slouží zejména pro vyšší rozdíly tlaků a základní oddělení dvou systémů. Propojení je do pravého úhlu, jak je možné vidět na Obr. 1.10.
- Rovný 180 – podobný ventilu úhlovému, ale propojuje v přímce.
- Motýlkový – využívá se pro regulaci přítoku plynu k pumpě. Zabírá málo místa. Tzv. *gate valve* se využívá pro podobné účely.
- *Ball Cock* – je určen zejména k oddělení dvou systémů a nabývá pouze dvou stavů.
- Jehlový ventil – Obvykle jej nelze plně dovřít, ale jinak poskytuje libovolnou pozici. Slouží především pro vpouštění pracovního média do aparatury. Pro plné zavření je třeba jej kombinovat s dalším ventilem.
- Membránové – jedná se o velmi kompaktní ventily, které jsou schopné měřit průtok plynu.
- Dávkovací ventil – jak již název napovídá, slouží pro jemné ovládání průtoku.

Ve vakuových aparaturách se ventilů využívá jako jednoho ze dvou hlavních regulačních prvků (společně s vývěvou). Volí se konkrétní ventil podle velikosti dané aparatury a tedy požadovaného průtoku, rychlosti operace ventilu a také jeho přesnosti. Pro nejpřesnější aplikace se využívá hmotnostních průtokoměrů často označovaných také jako MFC, což je zkratka pro anglické *mass flow controller*. [17]



Obr. 1.10: Diagram úhlového ventilu [17]

## 2 Analogově-digitální převodníky

Analogově digitální převodníky (ADC – Analog-Digital Converter) zajišťují propojení mezi spojitym a číslicovým prostorem. Konkrétně se zabývají převodem vstupního napětového signálu na digitální kód  $n$ -bitů, jejichž počet odpovídá rozlišení převodníku. Inverzním procesem k tomuto převodu se potom zabývá zařízení známé jako digitálně analogový převodník (DAC – Digital-Analog Converter). Proces převodu spojitého signálu na číslicový se dá rozdělit do následujících úkonů:

- Vzorkování (*sampling*) – jedná se o proces zachycení a zapamatování aktuální hodnoty vstupního napětí v konkrétní čas, který je dán vzorkovací periodou převodníku. Zapamatování musí vydržet po dobu trvání této periody, během níž převodník hodnotu dále zpracovává. V blokových schématech se vzorkovací blok často označuje zkratkou „S/H“, která vychází z anglického *sample and hold*.
- Kvantizace (*quantization*) – během tohoto úkonu se převádí napětová hodnota na digitální kód o  $n$  bitech. Proces do výsledku zanáší tzv. *kvantizační chybu* danou změnou napětí, která reprezentuje změnu nejméně významného bitu (LSB – least significant bit). Převodník totiž nezaznamená menší změnu napětí a kvantizační chyba je tak dána touto rozlišovací schopností.
- Odeslání zpracované informace – zatímco původní řešení převodníků odesílá informaci po několika paralelně zapojených kabelech, kdy každý z nich reprezentuje jeden bit výsledku, v moderním provedení se nejčastěji implementuje některý ze sériových protokolů jako je I<sup>2</sup>C nebo SPI. V některých náročných aplikacích dochází i k zavedení komplexnějších průmyslových protokolů.

Dělení analogově-digitálních převodníků se obvykle stanovuje zejména podle toho, jakým způsobem realizují první dva zmíněné úkony.

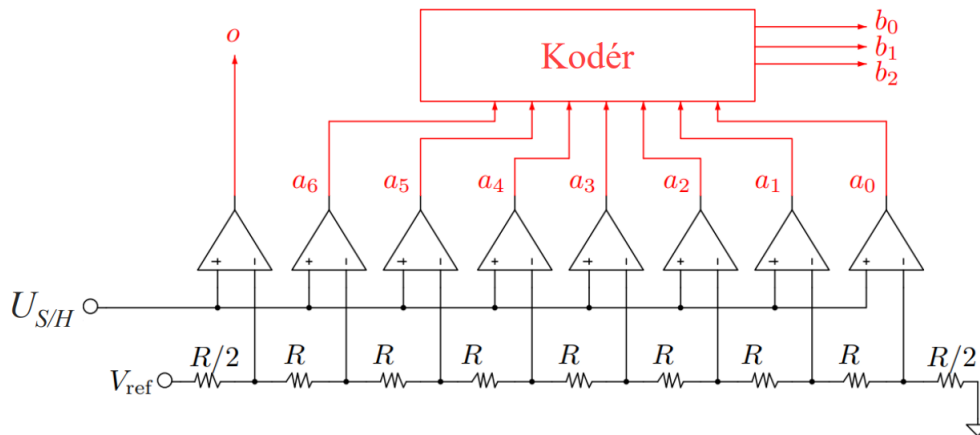
Rozdělení převodníku podle jejich rozlišení souvisí s počtem bitů výsledku. Pokud je u daného převodníku uváděno rozlišení  $n$  bitů, znamená to, že výsledná hodnota může nabývat celkem  $2^n$  stavů. Ze znalosti rozlišení a maximálního napětového rozsahu, který je převodník schopen zpracovat, můžeme stanovit tzv. *kvantizační krok*.

Ten představuje onu nejmenší možnou změnu vstupního signálu, kterou je převodník schopen promítnout do výsledku. Kvantizační chyba je potom menší nebo rovna polovině tohoto kvantizačního kroku. Typické hodnoty počtu bitů převodníků se mohou pohybovat od jednotek až po desítky, což vede na velmi rozličné přesnosti.

Dalším důležitým faktorem, podle kterého lze převodníky rozdělit, je jejich povolený rozsah vstupního napětí. Ten přímo souvisí s napětovou referencí, kterou převodník využívá a se kterou srovnává převáděné napětí. Většinou se jedná o interně vytvářenou referenci, ale v aplikacích, které kladou důraz na vysokou přesnost



se potom využívá referencí externích. Dle povolené polaroty napětí, které je na vstup převodníku možné přivést, potom lze rozlišit mezi unipolárním nebo bipolárním převodníkem.



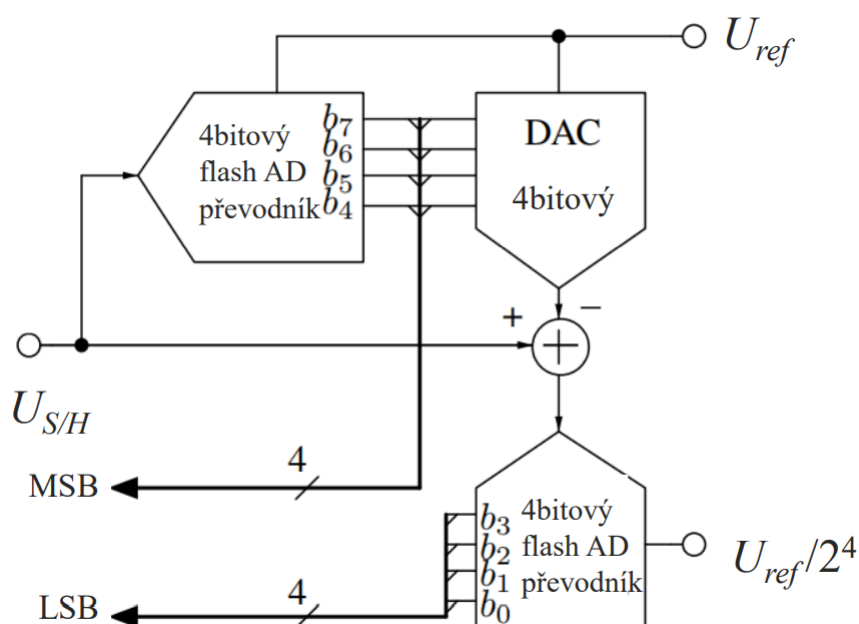
Obr. 2.1: 3bitový flash převodník.  $U_{S/H}$ : navzorkované napětí,  $U_{ref}$ : referenční napětí [9]

## 2.1 Flash převodníky

Nejdůležitějším je však rozdělení převodníků podle způsobu, jakým provádí převod napětí na digitální signál. Na velmi jednoduchém principu je založen převodníku typu *flash*, který vstupní napětí jednoduše srovná s několika referencemi a tu nejpodobnější z nich určí jako výsledek převodu. Realizuje se pomocí celé řady komparátorů, jak lze vidět na Obr. 2.1, které ono srovnání s referencí provádějí. Ta je pro každý z nich jiná a většinou se vytváří pomocí odporové sítě, která dělí celkové referenční napětí na menší části. Výstupy každého z komparátorů jsou potom zpracovány ve speciálním kodéru, který tyto hodnoty převede na správně kódované binární číslo. Vzhledem k tomu, že komparátory srovnávají vstupní napětovou hodnotu všechny současně, je převod velmi rychlý. Značnou nevýhodou těchto převodníků je však potřeba značného množství komparátorů a fyzická velikost odporové sítě. Počet celkově použitých komparátorů totiž odpovídá počtu možných výstupů převodníku. Odporová síť zase pro převodníky s vyšším rozlišením zabírá většinu plochy jejich čipu. Obvykle se tedy tyto flash převodníky realizují pouze pro menší počet bitů a jsou tak poměrně nepřesné. Možným řešením na principu rozdělení a panuj (*divide and conquer*) představují tzv. *half-flash ADC*, jejichž příklad lze nalézt na Obr. 2.2.

## 2.2 Half-flash převodníky

Half-flash převodníky využívají několik dílčích ADC i DAC pro zhotovení přesnějšího výsledku v porovnání s čistým flash převodníkem. Jejich obecný princip se dá popsat zhruba takto: první převodník typu flash provede zpracování vstupního signálu a na svém výstupu tak nabídne přibližnou hodnotu výsledku. Ta je přivedena na vstup prvního digitálně-analogového převodníku, který vrátí napětí o takové úrovni, která odpovídá digitálnímu výstupu prvního AD převodníku. Rozdíl mezi navzorkovanou hodnotou vstupního napětí a výstupní hodnotou prvního DAC je posléze přiveden na vstup druhého flash AD převodníku, který porovnává již pouze tuto rozdílovou hodnotu a zlepšuje tak celkovou přesnost výsledku.



Obr. 2.2: Blokový diagram 8bitového half-flash převodníku [9]

Oproti klasickému flash převodníku tímto postupným převodem dochází sice k určitému poklesu rychlosti, ale většinou se s výměnou za výrazně vyšší přesnost spokojíme.

## 2.3 Převodníky s postupnou aproximací

Převodníky s postupnou aproximací (*successive approximation*) implementují speciální registr SAR – *successive approximation register*, který slouží k postupnému vytváření (odhadování) výsledné hodnoty. Poté co tento registr provede odhad a

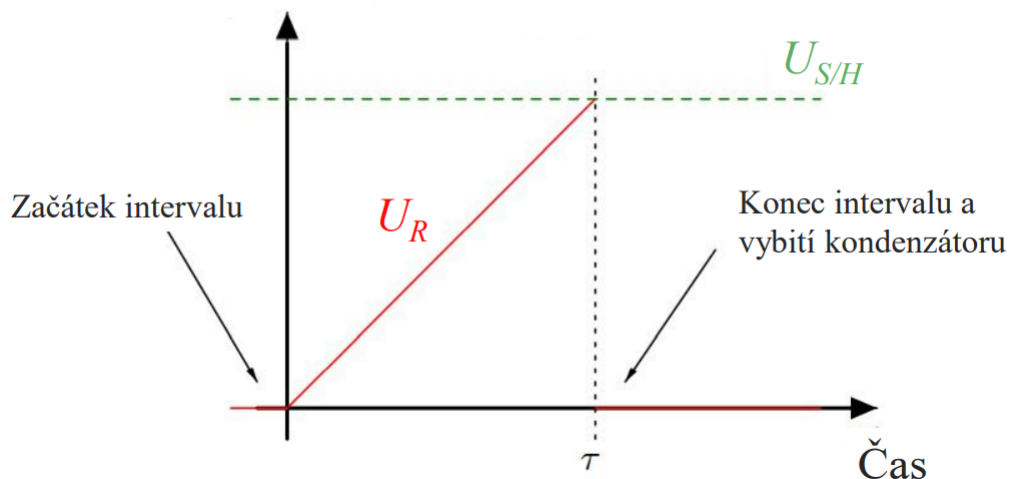
vytvoří jeho digitální hodnotu, přivede se tato na vstup digitálně analogového převodníku. Hodnota napětí na jeho výstupu se dále srovná s hodnotou navzorkovanou a podle výsledku tohoto srovnání se stanoví hodnota tohoto bitu a pokračuje se dalším odhadem. Konkrétní postup vytváření tohoto odhadu je následovný: v registru postupné aproximace jsou procházeny bity od nejdůležitějšího po ten nejméně významný a tyto jsou po jednom přiváděny do binární hodnoty jedna. Po každé této dílčí změně je výstupní napětí DAC přivedeno na komparátor, kde je srovnáno proti napětí vstupnímu. Podle výsledku této komparace je daný bit registru ponechán nebo invertován na opačnou (nulovou) hodnotu. Po průchodu všech bitů registru je operace ukončena a obsah registru představuje výslednou hodnotu převodu.

## 2.4 Převodníky s rampou

Převodníky s jednoduchou nebo dvojitou rampou využívají převedení problému měření napětí na obecně dobře zvládané měření časového intervalu. Tyto převodníky se využívají u měření s požadavkem na vysokou přesnost, kde ale tolik nehraje roli rychlost převodu. V dnešní době jsou ale postupně nahrazovány sigma-delta převodníky, které budou probrány později. U převodníků s rampou využíváme referenčního napětí k vytvoření jakési lineární funkce tím, že jej připojíme na vstup integrátoru s možným resetem (vybitím). Ten je obvykle realizován pomocí operačního zesilovače a pasivních RC prvků. Díky této linearitě je velikost časového intervalu od začátku rampy po okamžik, kdy se aktuální hodnota napětí rampy rovná vstupnímu navzorkovanému napětí, úměrná právě vstupnímu napětí, jak ukazuje Obr. 2.3. Tento princip však předpokládá perfektní znalost hodnot pasivních součástí. Ty se ale časem mohou od stavu z výroby lišit (driftovat) a do výsledného převodu se tak zanáší chyba. Možným řešením jsou převodníky s tzv. dvojitou rampou, kde dochází k opětovnému nabíjení a vybíjení kondenzátoru za konkrétních časových intervalů a díky principu vnitřní recalibrace potom nemá fluktuace jeho kapacity na výsledek převodu vliv.

## 2.5 Převodník $\Sigma\Delta$

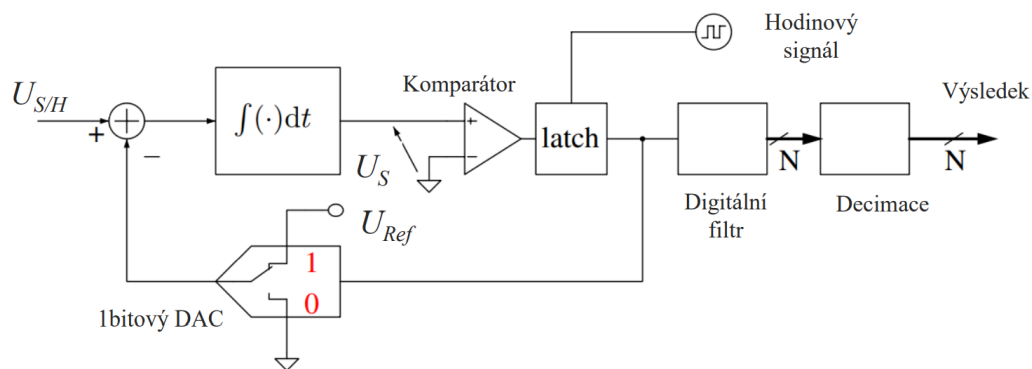
Název Sigma-Delta převodníků vychází z operací, které jsou označovány písmeny obsaženými v názvu tohoto typu ADC. Jedná se o integraci (suma  $\Sigma$ ) a porovnání (rozdíl dvou hodnot  $\Delta$ ). Při převodu dochází k opakovanému porovnávání výstupu integrátoru s referencí a díky zavedení zpětné vazby se výsledek porovnání projeví zpět na vstupu integrátoru. Konkrétní princip vytváření digitální hodnoty je



Obr. 2.3: Princip stanovení výsledku převodu z časového intervalu u převodníků s rampou [9]

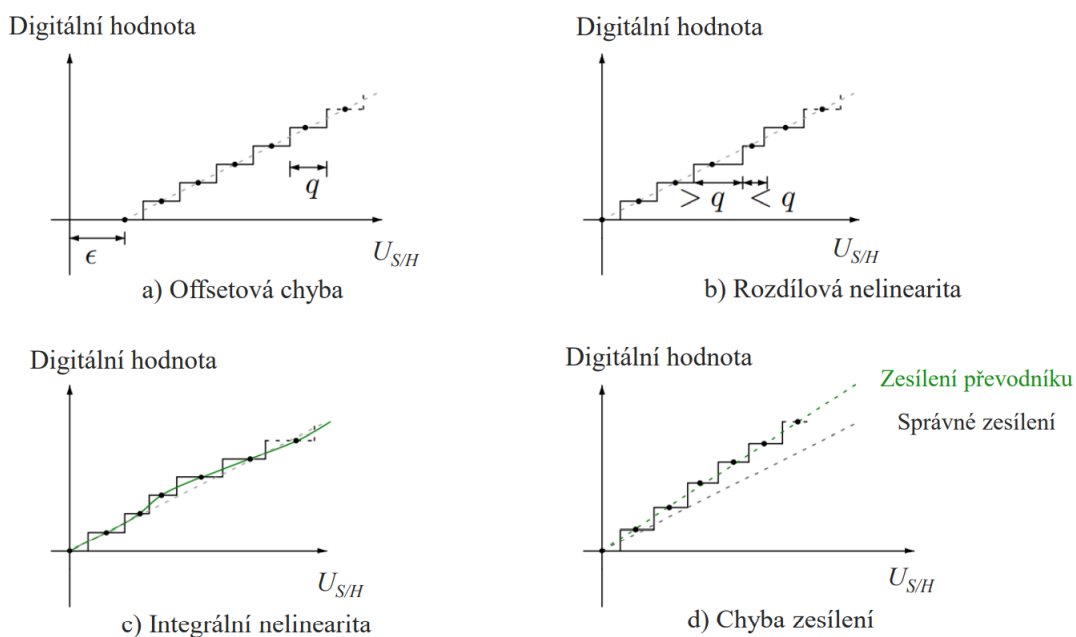
takovýto: navzorkované napětí je přivedeno na vstup integrátoru, jehož výstup je přiveden na komparátor. V případě přivedení kladného napětí na vstup se integruje tak dlouho, dokud se výstup komparátoru, na jehož invertující vstup je přivedena jakási napěťová reference, nepřeklopí do kladné hodnoty odpovídající logické úrovni jedna. Výstup tohoto komparátoru je s každým hodinovým cyklem zaznamenáván v podřípném registru. Digitálně analogový převodník v zavedené zpětné vazbě v případě přivedení logické jedničky sepne k referenčnímu napětí a na vstupu integrátoru se tak objeví záporná napěťová hodnota daná rozdílem navzorkovaného a referenčního napětí. Tím začne docházet k postupnému poklesu výstupní hodnoty integrátoru. Pokud je původní navzorkovaná hodnota napětí velká, zabere pokles až k záporným hodnotám několik hodinových cyklů, pokud je naopak hodnota nižší, pokles bude kratší. Počet jedniček zachycených na výstupu komparátoru s každým hodinovým cyklem je tak úměrný velikosti navzorkovaného napětí a počet nul je zase úměrný rozdílu referenčního a navzorkovaného napětí. Blokové schéma takového převodníku je uvedeno na Obr. 2.4. Pro zvýšení vzorkovací frekvence a dosažení vyšší přesnosti převodu existují různé speciální úpravy tohoto převodníku v jednoduché realizaci. Zásadní je také zmínit, že v dnešní době jsou sigma-delta převodníky nejrozšířenějším typem ADC vůbec, a to zejména díky své relativní univerzálnosti.

Důležité je v této části uvést i některé z chyb, kterých se převodníky při vytváření výsledné digitální hodnoty dopouštějí. Tzv. *chyby offsetu* (na Obr. 2.5 vlevo nahoře) jsou takové statické chyby, které způsobují posunutí všech výstupních hodnot o nějakou konstantu. *Chyba rozdílové nelinearity* (na Obr. 2.5 vpravo nahoře) se objevuje, pokud není kvantizační krok pro všechny úrovně vstupního napětí totožný.



Obr. 2.4: Základní blokové schéma sigma-delta převodníku [9]

Dochází potom k jakémusi nerovnoměrnému skoku výstupu určitých napěťových hodnot. Akumulací těchto rozdílových nelinearit vzniká *chyba integrální nelinearity* zobrazená na Obr. 2.5 vlevo dole. Zásadní je také uvést *chybu zesílení* (na Obr. 2.5 vpravo dole), která má za vliv rozdílnost směrnice výsledné charakteristiky převodu od hodnoty ideální. [9]



Obr. 2.5: Znázornění typických chyb převodu [9]

Převodník použitý pro tuto aplikaci bude sloužit k následujícím účelům: napěťový výstup vakuových měrek převede na digitální číslo dostatečně přesně, a to potom odešle k dalšímu zpracování do řídicí jednotky. Převodník by měl být určený pro všeobecnou aplikaci a umožňovat diferenciální měření, protože značná část měrek poskytuje z principu své funkce právě diferenciální výstup. Byť tomu tak v praxi

nemusí vždy být, samotní výrobci použitých měrek (jak bude rozebráno v následující kapitole) diferenciální měření doporučují. [26] Povolený vstupní napěťový rozsah nehraje příliš roli, protože výstupní napětí měrek lze pohodlně rozdělit pomocí odporového děliče, ale měl by být přibližně v řádu voltů, aby se eliminoval vliv šumu. Pro tuto aplikaci je zásadnější rozlišení převodníku než jeho rychlost. Čerpání vakuového systému je relativně časově náročný proces a dynamika systému tedy nebude příliš rychlá a změny tlaku tak není nutné zaznamenávat s vysokou rychlostí. Rozlišení je třeba zvolit takové, aby chyba výsledku způsobena převodem analogové hodnoty na digitální byla zanedbatelná. Vzhledem k obvyklým přesnostem měrek, jejichž chyby měření se nezřídka pohybují v jednotkách (někdy i desítkách) procent, bude dostačující zvolit převodník s rozlišením zhruba 14 bitů nebo více. Takový převodník je schopen rozlišit již  $2^{14} = 16384$  hodnot a například při napěťovém rozsahu 10 V je tak kvantizační krok roven napětí méně než jednomu milivoltu. Převodník by měl také disponovat alespoň dvěma kanály, na kterých je možné provádět paralelní měření. To umožňuje využití dvou měrek pro snížení vlivu jejich poruchy na řízení systému.

Dle těchto požadavků byl zvolen převodník typu MCP3423, který poskytuje dvoukanálové diferenciální měření s rozlišením 18bitů. [22] V době výběru tohoto převodníku hrála značnou roli jeho dostupnost na jinak omezeném trhu integrovaných čipů. Dalším vhodným kandidátem by mohl být převodník typu ADS131, který v podobné cenové kategorii nabízí lepší vlastnosti. [33]

### 3 Návrh řízení systému

Pro návrh systému řízení tlaku ve vakuové aparatuře je možné vycházet ze dvou zdrojů. Zaprvé se jedná o kusy techniky, které jsou předem dané a zajištěné – zde se jedná zejména o dražší části vybavení vybrané odborníkem jako vakuová vývěva nebo použité měrky, a zadruhé jde o ty části systému, které dle vlastního uvážení mohu sám zvolit. V první kategorii, tedy vybavení, které k realizaci zadání budu využívat, ale ne sám volit, se nachází veškerá vakuová technika. Konkrétně se jedná o dvou-  
stupňové zapojení vakuových vývěv, které slouží pro odčerpávání plynu z vakuové aparatury, vakuové měrky zaznamenávající aktuální hodnotu tlaku v aparatuře a také ovladatelný ventil, který se dle pokynů operátora přivírá nebo otevírá. Jedná se tak o poměrně standardní sadu nástrojů pro řízení tlaku ve vakuové aparatuře. O zmiňovaných zařízeních jsou známy následující informace:

- Vakuová pumpa – Jedná se o dvojici vývěv. První slouží pro připravení správného pracovního tlaku pro vývěvu druhou a můžeme ji tak nazývat podpůrným čerpadlem. Tato pumpa po spuštění již nevyžaduje žádný další operátorský zásah. Druhou vývěvou je vývěva turbomolekulární, která je již určena k regulaci tlaku v aparatuře. Zde se bude jednat o vývěvu značky Pfeiffer a bude v provedení dvou zařízení – samotná pumpa a k ní připojený ovladač, který dle externích pokynů řídí její chování a zajišťuje napájení. Pro mě je tak v tuto chvíli více důležitý tento ovladač než použitá vývěva, jelikož skrz ten je možné zařízení povelovat. Jedná se o ovladač typu TC 110, který umožňuje připojení k celé řadě vývěv. [27] Společnost Pfeiffer ve svých zařízeních využívá speciálního protokolu s názvem *Pfeiffer Vacuum Protocol*, který udává podobu dat, kterými je možné tato zařízení povelovat. Tato data jsou ovladači vývěvy posílána po sériové lince RS-485. Napájení vývěvy pak probíhá, jak již bylo uvedeno, také přes připojený ovladač a to z napěťové úrovně 24 V. Napájecí proudy se odvíjí od příkonu zvolené pumpy.
- Vakuové měrky – I v tomto případě se jedná o produkt firmy Pfeiffer, konkrétně je potom při běžné operaci zařízení využívána kombinovaná měrka PKR-251. [26] Ta produkuje napěťový signál v rozsahu přibližně 0-10 V jako výsledek svého měření. Pro napájení těchto měrek je třeba napěťový zdroj o hodnotě přibližně 15-30 V a spotřeba činí maximálně 2 W. Funkcionalita, kterou užívaná měrka nabízí, umožňuje její identifikaci řídicím zařízením pomocí změření velikosti integrovaného identifikačního odporu.
- Ventil – opět zvolen z portfolia společnosti Pfeiffer. Jedná se o typ EVR 116, který umožňuje ovládání hned dvěma způsoby. Buď pomocí přivedení napětí na analogové vstupy ventilu anebo pomocí jednoduchého sériového rozhraní. To umožňuje posílat příkazy a získávat z ventilu informace podobně jako pomocí

*Pfeiffer Vacuum Protocolu*, ale v alternativním a méně komplexním provedení. Ventil je opět napájen z jednotného napětí 24 V a maximálně pro svůj chod potřebuje až 0,5 A dodávaného proudu. Jeho maximální proudový odběr tak činí 12 W.

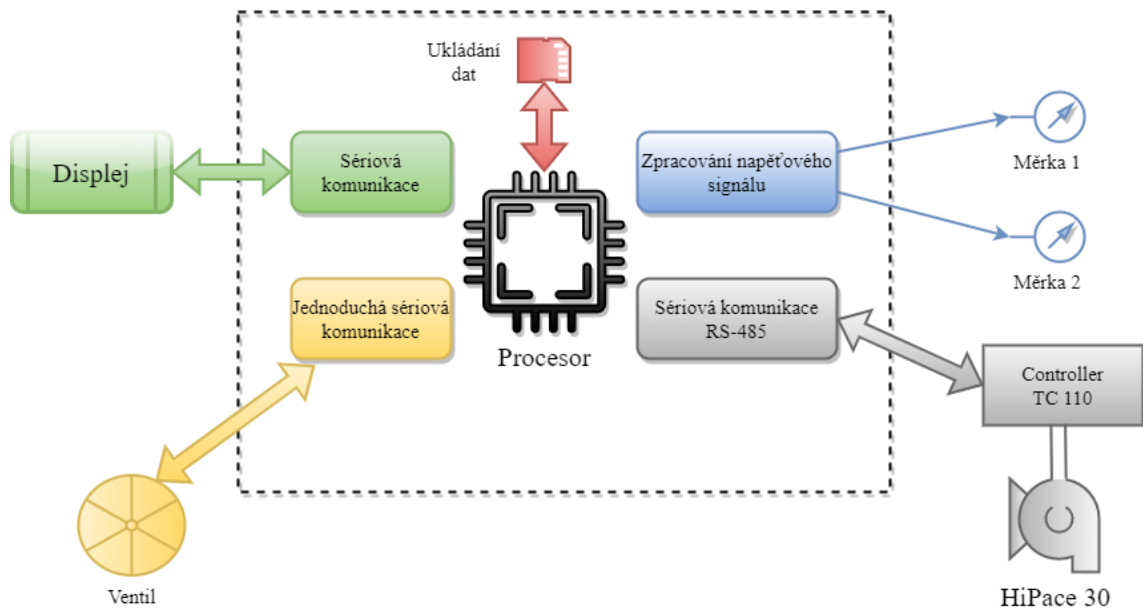
Druhá kategorie zařízení, která budou potřeba pro sestavení systému pro ovládání tlaku ve vakuové aparatuře, obsahuje zejména tyto dvě: řídicí jednotku, která zajišťuje ovládání a napájení všech vakuových komponent uvedených výše a také zobrazovací jednotku (dále displej), která zase umožňuje povelování systému ze strany operátora. Řídicí jednotka by měla umožňovat jednoduché připojení všech komponent a zbavit obsluhu nastavování jednotlivých zařízení tak, aby byla schopna provozu. Bude tedy třeba, aby umožňovala paralelní komunikaci po několika kanálech: po sériové lince RS-485 pro ovládání turbomolekulární vývěvy, jednoduché sériové rozhraní pro povelování ventilu a také musí umět zpracovávat analogové napěťové signály, které generují měrky. Všechny tři způsoby komunikace musí centrální jednotka obsluhovat zároveň a v další části realizace musí být také schopna na jejich základě ventil a pumpu ovládat tak, aby v aparatuře došlo k ustálení požadovaného tlaku.

Paralelně k této činnosti musí být jednotka také schopna data ze všech zařízení přehledně zobrazovat na připojeném displeji. V neposlední řadě také musí být schopna zpracovávat z tohoto displeje pokyny uživatele a dle nich se zachovat, což představuje čtvrtý způsob komunikace, který řídicí jednotka obstarává. Pokud budeme hovořit o napájení, což by měla pro všechny moduly poskytovat, kromě sebe samotné a připojeného displeje (odhadovaná spotřeba do 5 W) je třeba počítat s těmito výkony: ventil při svém maximálním odběru vyžaduje příkon 12 W a měrky až  $2 \times W$ , jelikož zařízení má umožňovat práci se dvěma měrkami paralelně. V případě turbomolekulární vývěvy se potom jedná o zmíněnou kombinaci ovladače TC110 a libovolné s ním kompatibilní vývěvy, jejíž maximální proudová spotřeba při napájecím napětí 24 V nepřekračuje přibližně 5 A, což je projektovaná zatížitelnost konstruovaných komponent. Může se tak jednat například o pumpy HiPace 10 [27], HiPace 30 [3], či HiPace 80 [4] anebo další podobné splňující tyto požadavky. Při maximálním možném zatížení se tak ze strany vývěvy jedná o potřebný dodávaný výkon přibližně 120 W. Celkově je tak včetně rezervy třeba počítat s výkonem, který přes navrhovanou řídicí jednotku bude nutné dodávat, rovnajícimu se přibližně 140–150 W.

Další roli, kterou by zhotovené zařízení mělo plnit, je ukládání hodnot z měření, která na něm probíhají. K tomuto účelu musí řídicí jednotka obsahovat modul, který tuto činnost umožňuje. Jedná se tak o další komunikaci, kterou řídicí jednotka obsluhuje, ale k tomuto nemusí docházet v krátkém čase a paralelně s obsluhou ostatních zařízení.



Tyto slovně rozepsané požadavky jsem shrnul pomocí diagramu znázorňující celý systém na Obr. 3.1.



Obr. 3.1: Blokové schéma systému pro řízení tlaku ve vakuové aparatuře

## 4 Řídící jednotka

Řídící jednotka musí sloužit jako výkonný ústřední člen celého zařízení, být spolehlivá a její programování nesmí představovat příliš náročný úkol, ať už čistě po stránce složitosti, nebo například tím, že svou nadměrnou komplexností do systému zanáší celou řadu nových problémů, které je třeba řešit. Konkrétně bychom potom požadavky na řídicí jednotku mohli vyjmenovat takto:

- Výkonnost – musí být dostatečně výkonná, aby zvládala plnit všechny funkce definované v předchozím bodě dostatečně rychle. Konkrétní náročnost těchto úloh je však obtížné dopředu přesně stanovit. . . .
- Jednoduchost programování – řídicí jednotka by měla být jednoduše programovatelná i relativním začátečníkem v oboru, aby tento proces zbytečně nezpomaloval celý vývoj. Dobrá dostupnost výukových materiálů je zde značnou výhodou.
- Spolehlivost/robustnost – tyto faktory lze snadno stanovit z recenzí uživatelů a posouzením zapojení dané řídicí jednotky do jiných projektů podobného typu. Pokud se jednotka již jinde v praxi osvědčila, je důvodné je považovat za spolehlivou.
- Dostatečné možnosti IO – vzhledem k požadavkům, které bude muset řídicí jednotka obsloužit z různých připojených periférií, je třeba, aby disponovala jednoduchou dostupností a širokou škálou vstupů a výstupů
- Rozměry a spotřeba – tyto parametry nejsou v naší laboratorní aplikaci příliš zásadní. Kompaktnost v této fázi návrhu nehraje roli a spotřeba vzhledem k napájení zařízení z elektrické sítě není důležitá. Faktory tak ve výběru vhodné řídicí jednotky nebudou zohledňovány.
- Doplnkový hardware – snadná dostupnost hardwarových řešení, které lze k centrální jednotce snadno připojit a rozšířit tak její funkcionalitu.

Vzhledem k tomu, jak často a s jakým úspěchem se v mém okolí vyskytují projekty využívající některou z desek Arduino, ať už na hobby nebo poloprofesionální úrovni, první volba řídicí jednotky byla jasná. Nyní uvedu, jaké konkrétní desky tato firma nabízí a do jaké míry splňují požadavky na řídicí jednotku uvedené výše.

Tab. 4.1: Shrnutí vlastností desek Arduino [6]

TYP	CPU	Provozní napětí	Velikost paměti	IO
Uno	16 MHz ATmega328P	5 V	32 KB, 2 KB SRAM, 1 KB EEPROM	14 – 6 AI, 6 PWM, 1 UART
Nano	16 MHz ATmega328	5 V	32 KB, 2 KB SRAM, 1 KB EEPROM	22 – 8 AI, 6 PWM, 1 UART
Due	84 MHz SAM3X8E ARM	3,3 V	512 KB, 64 + 32 KB SRAM	54 – 12 AI, 12 PWM, 4 UART
Mega	16 MHz ATmega2560	5 V	256 KB, 8 KB SRAM, 4 KB EEPROM	54 – 16 AI, 15 PWM, 4 UART

V následující pasáži textu popisují zásadní rozdíly mezi jednotlivými deskami. Tři z uvedených Arduino desek jsou poháněny 16MHz procesorem ATmega. Rozdíl mezi menším z nich (328) a větším (2560) je především v počtu vstupně-výstupních periférií, které je možné k procesoru připojit. To se promítá i do výsledného počtu vstupů a výstupů celé desky. Hnací jednotkou poslední z desek (Due) je potom ARM procesor s výrazně větším výkonem, který na rozdíl od předchozích tří, které pracují při napětí 5 V, operuje na nižší napěťové úrovni 3,3 V. Byť se na první pohled může zdát, že výkonnější procesor je jednoznačná volba, není tomu tak. U Arduino Due se často vyskytují softwarové nekompatibility a obecná podpora pro tuto desku není na stejné úrovni, jako u ostatních. [19]

Všechny z desek se programují pomocí základního Arduino IDE, případně pomocí IDE dle preferencí s doinstalovaným plug-inem pro programování desek Arduino v jazyce C++ s drobnými úpravami.

V případě porovnání desek na základě jejich vstupů a výstupů je výsledek jednoznačný. Desky Mega a Due disponují výrazně větším množstvím vstupních a výstupních portů a vzhledem k množství zařízení, které budou se zvolenou ústřední jednotkou komunikovat, se jeví jako lepší volby.

Paměť je na Arduino deskách obvykle dělena do až tří kategorií – Flash, SRAM a EEPROM. Paměť typu Flash slouží k uchování programu a inicializačních hodnot. Nelze do ní při běhu programu zapisovat a měnit tak data v ní uložená. K tomu je určena další z pamětí – SRAM. Ta ukládá jak data statických a globálních proměnných, tak vytváří paměť typu heap a stack, které známe z programování v C++. Posledním typem paměti, která se na Arduino deskách využívá, je EEPROM. Slouží k uchování malého množství dat, která jsou ale v paměti uložena i po výpadku napájení. Obvykle se jedná o stavy vstupů a podobně. [7]

Kondaveeti, Kumaravelu a Vanambathina [19] se v článku zaměřili na shrnutí vědeckých projektů, které ve svých řešeních implementovaly některé z desek Arduino. Ty se dle něj vyskytují i v projektech se zaměřením na automatizaci procesů, vývoj softwaru, mezihardwarovou komunikaci nebo se zaměřením všeobecným. Ve všech těchto aplikacích se desky Arduino osvědčily a vzhledem k podobnosti s řešeným problémem se dá předpokládat úspěšné nasazení některé z desek i v tomto projektu. Autoři jako zásadní výhody použití Arduino řešení uvádí kromě dobré dostupnosti a příznivé ceny jednoduchost práce s deskou nebo širokou uživatelskou základnu. Také se pouští do srovnání s dalšími podobnými produkty jako Raspberry Pie nebo WaspMote. Vzhledem k požadavku na jednoduchost a hardwarovou rozšiřitelnost však Arduino vychází pro tento projekt jako lepší řešení.

Další nespornou výhodou je značný výběr právě doplňkového hardwaru, který lze dokoupit a k desce díky standardizovaným rozměrům a osazeným kolíkovým lištám jednoduše připojit. Pro vývoj se tak jedná o zcela ideální řešení, které řídicí jednotku

doplní o nové funkce, které samotná jednotka nenabízí. Z internetových obchodů jsou k sehnání rozšiřující moduly pro většinu základních funkcí jako implementace SD karty, motorové karty anebo Wi-Fi. Jednoduchý je také proces návrhu vlastního hardwarového modulu, který tak systém rozšíří o nestandardní funkce konkrétní aplikace. Nový modul musí pouze zvolené řídicí jednotce odpovídat rozměrově a umístěním používaných pinů.

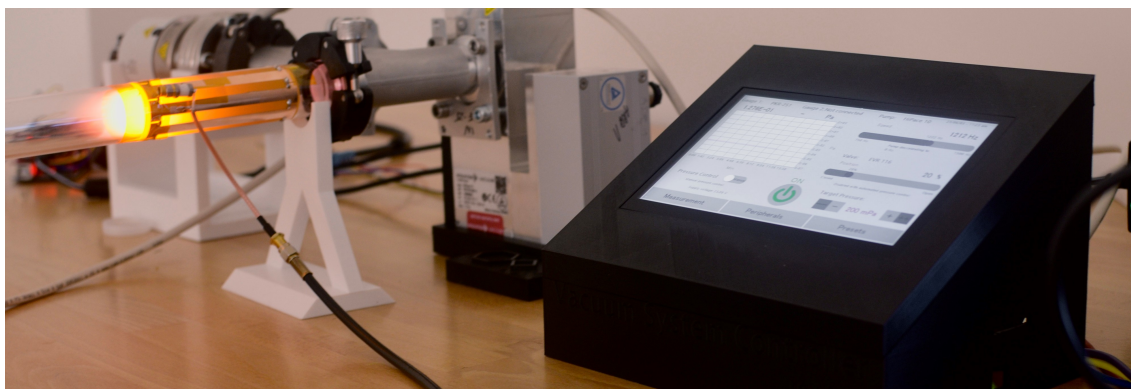
Nakonec představuje Arduino se svým IDE výhodu v podobě multiplatformnosti. Lze tak v jednotném prostředí zařízení programovat jak z počítače se systémem Windows, tak MacOS X nebo jedné z Linuxových distribucí.

Po zvážení všech zmiňovaných faktorů jsem pro realizaci projektu zvolil deska typu Arduino Mega a to z následujících důvodů: Oproti deskám Nano a UNO disponuje větší pamětí pro program i proměnné a také nabízí dostatečné množství vstupních a výstupních portů. Oproti desce Arduino Due zase disponuje větší softwarovou kompatibilitou a její užití je v podobných aplikacích lépe ověřené.

Pro realizaci uživatelského rozhraní, kterým bude operátor se zařízením moci jednoduše komunikovat, jsem na základě mnohých doporučení zvolil řešení od firmy Nextion. Ta nabízí dotykové (resistivní i kapacitní) displeje o různých velikostech, které jsou velmi jednoduché na používání. [24] Díky procesoru, který je na každém modelu přítomen, běží program uživatelského rozhraní přímo na displeji a nezatěžuje tak řídicí jednotku nedůležitými požadavky. Komunikace mezi displejem a jednotkou probíhá pouze v nutných případech, a to po sériové lince. Displeje jsou napájeny napětím 5 V, stejně jako provozní napětí zvolené Arduino desky Mega, což usnadňuje integraci displeje do systému. V první iteraci bude zvolen displej o kompaktní úhlopříčce 2,4 palce. Obrovskou výhodou řešení Nextion, a také důvod, proč jej v podobných projektech celá řada lidí doporučuje, je zdarma dostupný software Nextion Editor, který umožňuje tvorbu programů, které později na displeji běží. Návrh je tak velmi intuitivní a jednoduchý. Konkrétním zvoleným modelem je potom ten, který od výrobce nese označení NX8048P070-011C a má tyto parametry: rozlišení displeje je 800×480 pixelů při fyzické úhlopříčce 7", způsob registrace dotyku je kapacitní a takt procesoru, kterým je displej vybaven, je 200 MHz. Jedná se o produktovou řadu *Intelligent*, která nabízí ze všech modelů displejů Nextion nejvíce možností i nejvyšší výkon.

## 5 Popis systému

Pro potřeby měření a řízení tlaku ve vakuové aparatuře jsem dle teoretického návrhu sestrojil zařízení, pracovně označované jako *Vacuum System Controller*, které v kompaktním provedení poskytuje obsluze systému kontrolu a přehled nad probíhajícími měřeními a zabezpečuje plynulý provoz vakuové aparatury. Zařízení připojené k aparatuře, v níž tlak ovládá, je vidět na Obr. 5.1

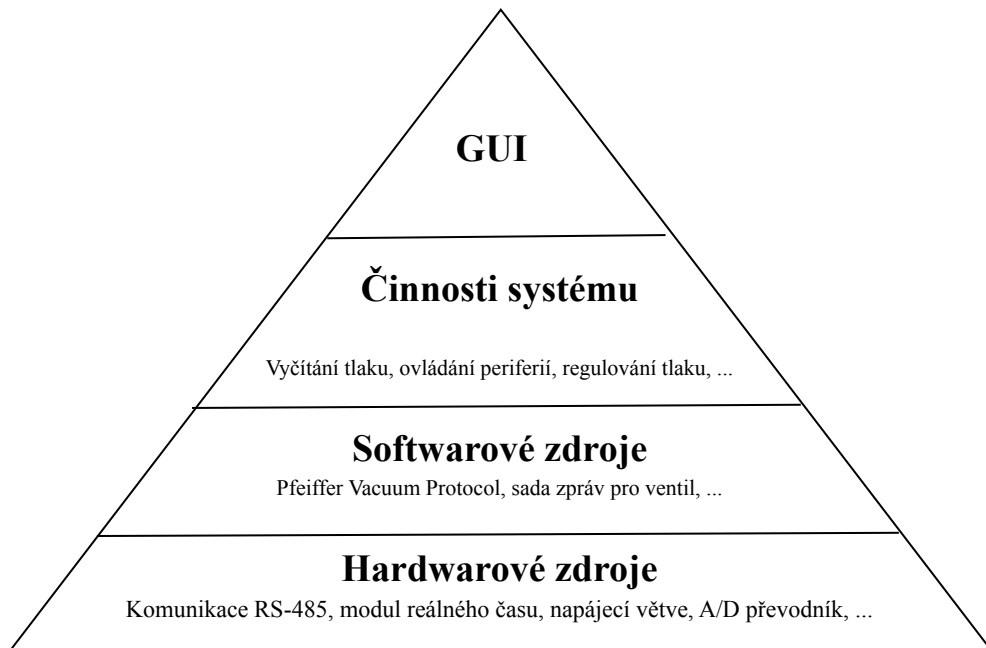


Obr. 5.1: Vacuum System Controller připojen k vakuové aparatuře

Mezi hlavní funkce zařízení patří komunikace a napájení připojených vakuových komponent tak, jak bylo navrženo v Kap. 3. Zařízení zabezpečuje mimo jiné povelování komponent dle pokynů obsluhy a vizualizaci aktuálního stavu systému. Zařízení *Vacuum System Controller* (dále VSC) jsem sestrojil jako skladný počítač, který je po zapojení k vakuové aparatuře vybavené kompatibilními komponentami schopný tyto komponenty identifikovat a dle pokynů obsluhy ihned zahájit činnost. K propojení VSC a aparatury slouží sada kabelů s konektory standardních typů, které poskytují jak spolehlivý přenos komunikačních signálů, tak zabezpečují tok dostatečně velkých proudů pro napájení komponent. Konektory kabelů zamezují špatnému zapojení a potenciálnímu poškození zařízení VSC i vakuových komponent. Možnosti všech podporovaných komponent jsem plně využil a sestrojil způsoby komunikace s nimi, které zajišťují kontrolu v maximální možné míře umožněné výrobcem.

Vizualizace aktuálně probíhajícího měření, stejně jako prvků sloužících pro intuitivní ovládání aparatury a periférií, probíhá na velkém dotykovém displeji zapuštěném do těla zařízení. Konkrétní podobu grafického rozhraní jsem vytvořil podle požadavků kolegů, kteří zařízení využívají pro konstruování plazmového zdroje určeného pro použití v iontovém motoru.

Pro ucelený popis zhotoveného zařízení a jeho provozních vlastností jsem zvolil následující přístup: nejprve popisují grafické rozhraní displeje – tedy tu část řešení, kterou uživatel jako první vidí a se kterou interaguje. Dále se zabývám vysvětlením

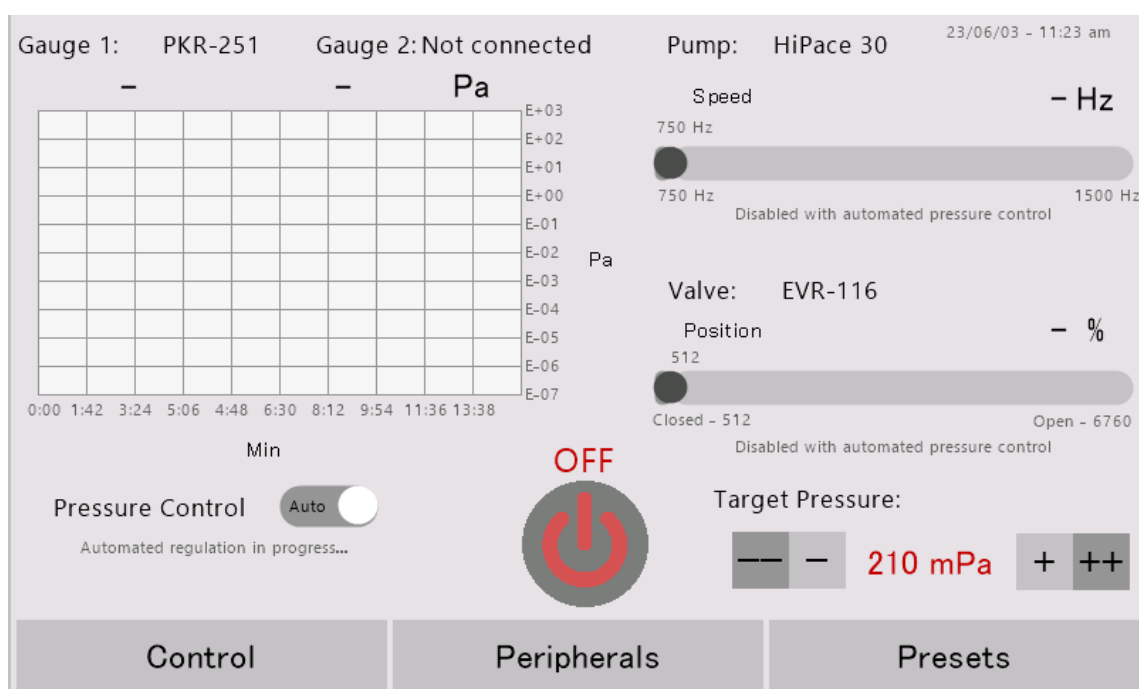


Obr. 5.2: Logická struktura popisu VSC

implementace funkcí, které se skrývají za jednotlivými tlačítky a jinými prvky tohoto uživatelského rozhraní. V této části tak nejdříve popisují hardwarovou stránku projektu, kde vysvětlují obvody, které jsem pro správnou činnost VSC implementoval, a jejich funkce, fyzická propojení a použité součástky. Nadstavbu nad těmito *hardwarovými zdroji* tvoří v další kapitole popisované zdroje softwarové, které jsem sám napsal či pod sdílenou licenci využil pro zprovoznění požadovaných funkcí VSC. Sem se řadí například knihovny napsané pro komunikaci s vakuovými periferiemi, rutina převodu napěťových hodnot na digitální čísla nebo zápis dat na SD kartu. Posledním stupněm popisu jsou potom činnosti systému, které pro své fungování využívají dostupných *softwarových zdrojů* a realizují díky nim složité a komplexní operace, kam patří zejména ovládání periferií nebo realizace regulátoru tlaku.

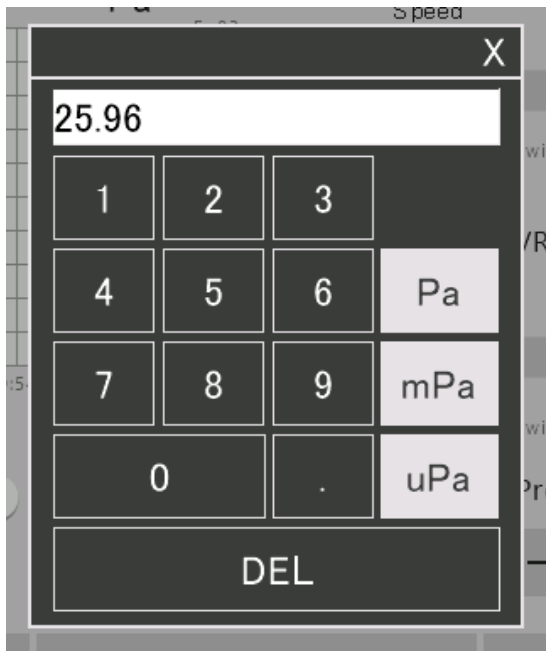
## 5.1 Uživatelské rozhraní

Tato část textu slouží jako popis navrženého rozhraní a zároveň krátký návod k jeho používání. Po připojení zařízení k napájení se po načtení a inicializaci systému zobrazí karta sloužící jako výchozí rozcestník pro přehled nad aparaturou. Jak je na Obr. 5.3 vidět, v dolní části se nachází tlačítka pro zobrazení dalších stránek rozhraní, v pravé části obrazovky jsou potom umístěny komponenty přímo (posuvníky rychlosti pumpy a polohy ventilu) nebo nepřímo (nastavení cílového tlaku) ovládající turbomolekulární vývěvu a ventil. V levé části se nachází kromě přepínače mezi manuálním a automatickým řízením aparatury i sekce s informacemi o aktuálních hodnotách tlaku a jejich historii vizualizované grafem.



Obr. 5.3: Uživatelské rozhraní – Hlavní stránka

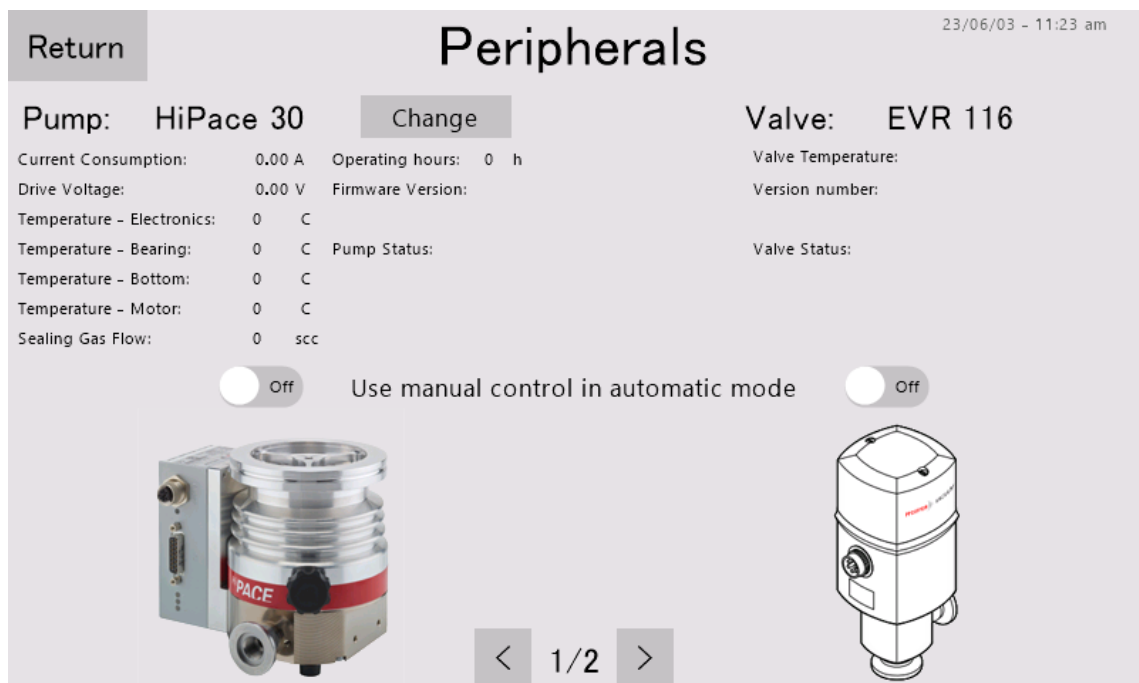
Nastavování rychlosti pumpy, polohy ventilu nebo požadovaného tlaku je možné činit buď pomocí prvků viditelných na Obr. 5.3, nebo po kliknutí na jejich příslušnou číselnou hodnotu pomocí následně zobrazené klávesnice, jejíž podoba pro nastavení tlaku je vidět na Obr. 5.4



Obr. 5.4: Uživatelské rozhraní – Klávesnice pro nastavení požadované hodnoty tlaku v aparatuře

Na karta obsahující nastavení vakuových periférií, která je zobrazena na Obr. 5.5, se dá přesunout z hlavní stránky stiskem tlačítka „Peripherals“. V první její části nalezneme informace o připojené turbomolekulární vývěvě a ventilu společně s obrázky ilustrující jejich podobu. Jedná se tak o interaktivní kontrolu pro obsluhu VSC, že jsou veškeré modely připojených periférií zvoleny správně. U vývěvy je také možnost připojený model změnit, podobně jako je tomu ve druhé části stránky s vakuovými perifériemi věnované měrkám. Uživatel buď může nechat systém pracovat automaticky a nechat jej připojené měřky identifikovat pomocí algoritmu popsaného v Sek. 7.3.1, nebo manuálně zvolit jejich model, jak ukazuje

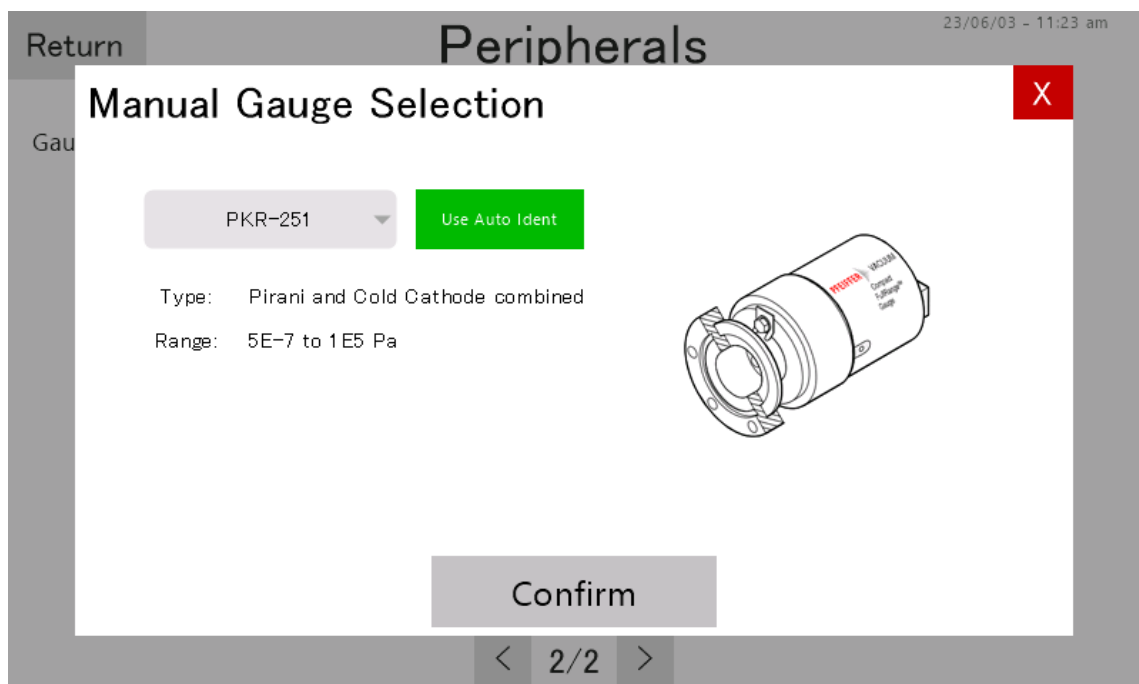
Obr. 5.6. Při volení se uživateli také zobrazují informace o rozsahu tlaků, který je s danou měrkou možné měřit a také o jejím typu (Pirani, Cold Cathode, apod.)



Obr. 5.5: Uživatelské rozhraní – Karta vakuových periférií (vývěvy a ventilu)

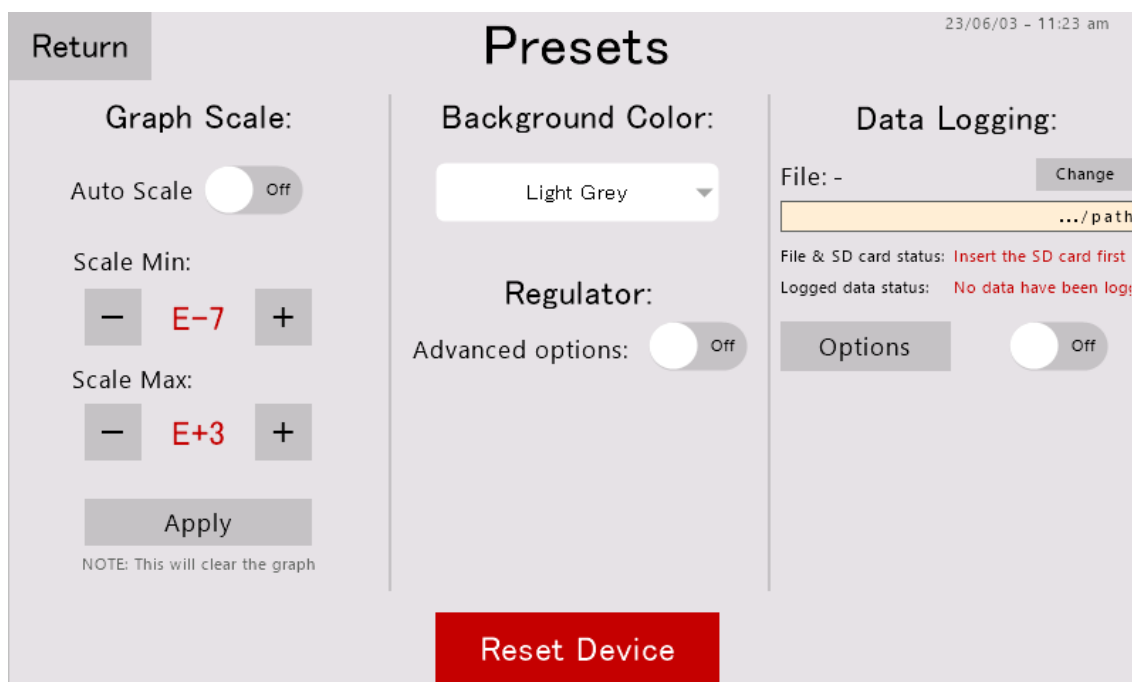


Karta uživatelských přednastavení je rozdělena na tři sloupce, jak je vidět na Obr. 5.7. V prvním sloupci lze měnit aktuální použité měřítko grafu minulých hodnot tlaku (na Obr. 5.3 vlevo) a v prostředním lze potom povolit pokročilé možnosti nastavení regulátoru, které více diskutuji v Sek. 8.3.3, a také měnit barvu pozadí uživatelského rozhraní podle preferencí obsluhy. V posledním sloupci karty „Presets“ lze nastavit ukládání měřených dat na vloženou kartu SD společně s názvem a umístěním souboru, do kterého se ukládání realizuje. Pod tlačítkem „options“ se potom skrývá nastavení možných hodnot, které se v každé periodě programu VSC mají ukládat, jako hodnoty tlaku, rychlost otáček vývěvy nebo poloha ventilu.

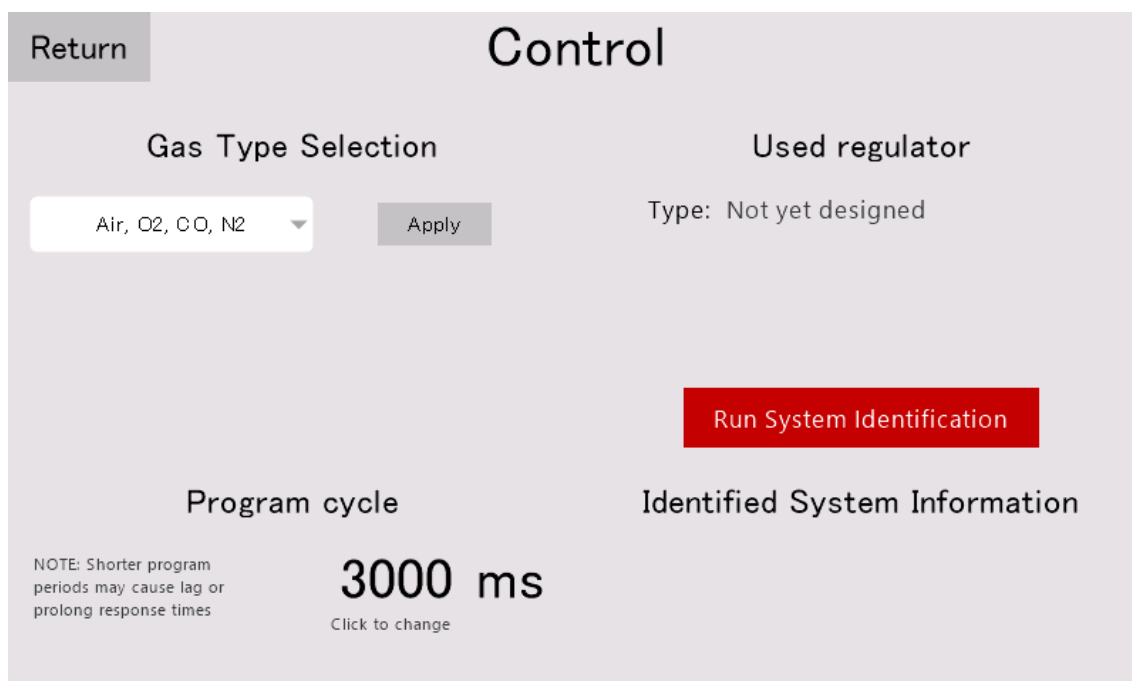


Obr. 5.6: Uživatelské rozhraní – Manuální volba připojené vakuové měřky

V kartě „control“ nalezneme možnosti řízení tlaku ve vakuové aparatuře. Pomocí tlačítka „Run System Identification“ je spouštěn proces automatické identifikace systému, který popisuji v Sek. 8.3.2. V levém horním rohu je pro budoucí implementaci připraven prvek pro změnu typu v aparatuře používaného plynu, což se projeví na rovnici závislosti tlaku na měřenosném napětí, jak ukazuje Obr. 7.4. Ve spodním levém rohu může obsluha volit požadovanou periodu opakování programové smyčky VSC.



Obr. 5.7: Uživatelské rozhraní – Karta uživatelských preferencí



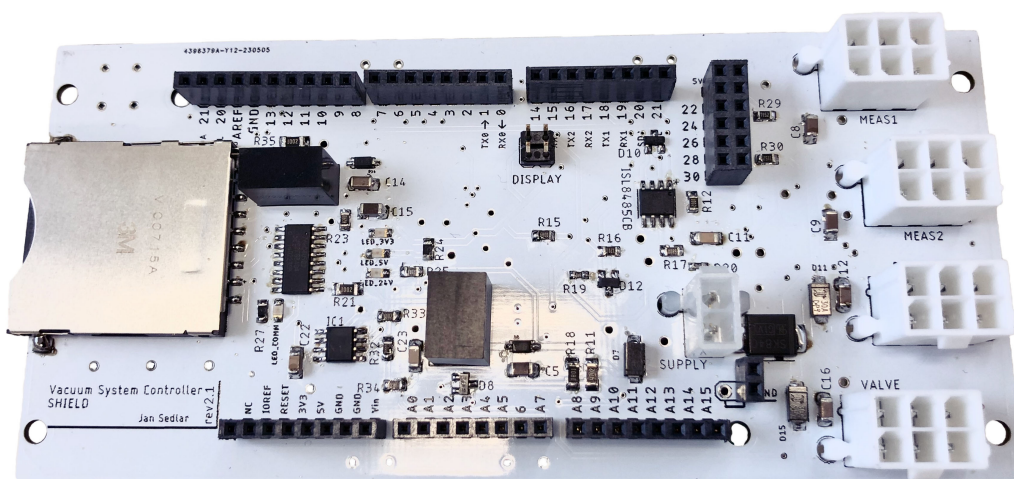
Obr. 5.8: Uživatelské rozhraní – Nastavení řízení vakuové aparatury

## 6 Hardwarové zdroje

V této kapitole jsou popsány hardwarové aspekty zařízení, jejich struktura, funkce a způsob jejich implementace.

### 6.1 Desky tištěných spojů a jejich propojení

VSC je interně realizováno jako základní řídicí jednotka zvolená v Kap. 4 a dvě na míru vytvořené desky tištěných spojů (PCB – printed circuit board) realizující funkce specifické pro tuto aplikaci. První z desek je konstrukčně realizována jako připojitelný *shield* (deska je svými piny zasunuta do řídicí jednotky a tyto piny kopírují její rozložení) a jsou na ní implementovány obvody realizující mnoho funkcí, zejména potom komunikaci s vakuovými periferiemi a displejem, modul uchování reálného času nebo slot pro SD kartu. Osazená deska je vidět na Obr. 6.1.

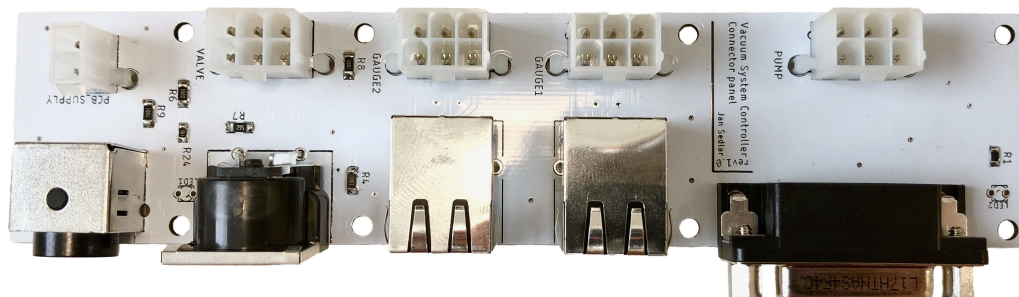


Obr. 6.1: Deska tištěných spojů s doplňkovými obvody

Druhá deska, která je v hotovém provedení vyobrazena na Obr. 6.2, slouží pro uchycení kabelových konektorů, které jsou vyvedeny na pouzdro přístroje a je propojena s první deskou pomocí robustních interních kabelů. Realizuje tedy primárně propojení vnitřně-vnější mezi řídicí jednotkou s shieldem a vakuovými periferiemi a také s napájecím zdrojem.

Deska s obvody je pro účely dalšího vývoje osazena průchozími lištami, které zajišťují jak propojení s řídicí jednotkou, tak umožňují případné připojení dalšího zásuvného modulu, či připojení sondy osciloskopu na libovolný pin zařízení. Toto řešení se v praxi pro vývoj zařízení osvědčilo, a protože vzhledem k velikosti displeje

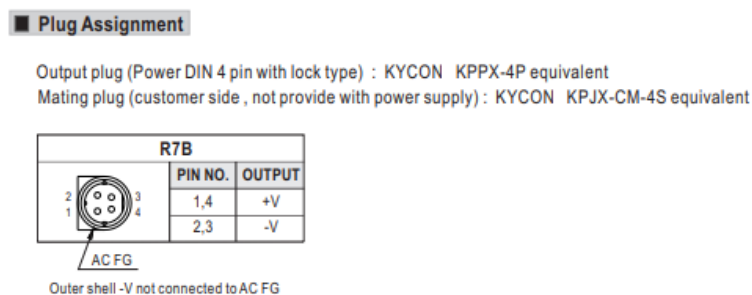
není v pouzdru zařízení nouze o místo, zůstalo zachováno i v této verzi desky. Interní propojení obou PCB je realizováno robustními konektory značky MOLEX [23] v provedení  $2 \times 3$  pro vakuové periferie nebo  $2 \times 1$  pro napájecí cestu.



Obr. 6.2: Konektorová deska tištěných spojů

## 6.2 Napájení

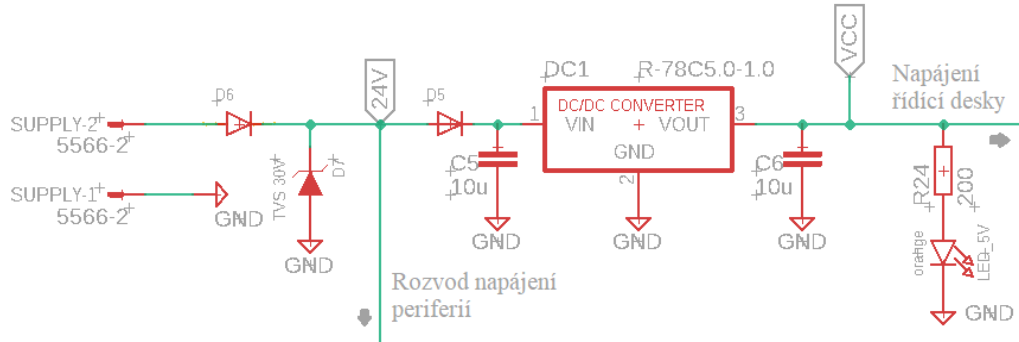
Zařízení je po připojení do elektrické sítě schopno samostatně operovat a slouží také jako primární zdroj energie pro všechny připojené periferie. Jak jsem zmínil v Kap. 3, pro napájení vakuových komponent je vhodné jednotné napětí 24 V, které také bylo zvoleno pro primární napájení systému. Pro připojení zdroje k zařízení jsem zvolil konektor KYCON KPJX-4S zobrazený na Obr. 6.3 [20], který umožňuje použití celé řady zdrojů vhodných pro tuto aplikaci – napájecí napětí 24 V a výkon 100–200 W Kap. 3, které využívají právě tohoto konektoru. [21]



Obr. 6.3: Konektor KPJX-4S z produktového listu vhodného napájecího zdroje GSM220A [21]

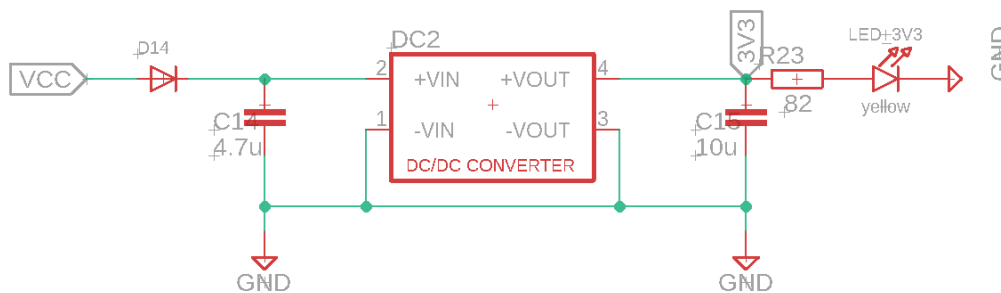
Pro interní propojení zase slouží zmiňovaný kabel MOLEX  $2 \times 1$ . Tato 24V napěťová úroveň je po desce dále rozvedena ke konektorům periferií a k DC-DC měničům,

kteřé vytváří napěťové úrovně vhodné pro integrované obvody a jinou drobnější elektroniku. Konkrétně jsou využívány standardní úrovně 5 V (viz Obr. 6.4) a 3,3 V (viz Obr. 6.5). První z nich slouží pro napájení řídicí desky (Arduino), displeje a valné většiny integrovaných obvodů. Nižší napěťová úroveň je potom využita v obvodech obsluhující připojenou SD kartu.



Obr. 6.4: Schéma zapojení vstupní ochrany a 5V DC-DC měniče

U obou rozvodných větví jsem také zapojil jednoduché SMD (surface mount device, opak THT – through hole technology; jedná se o způsob uchycení součástky přímo na pájivě plošky desky a nikoliv zapájení jejich vývodů do otvorů v desce) LED (light emitting diode), které indikují správnou funkčnost dané větve. Na fyzické desce jsou potom LED pro všechny přítomné napájecí úrovně umístěny na jednom místě a barevně odlišeny. Při správné funkci zařízení tak vidíme svítit trojici diod v červené, oranžové a žluté barvě. Zde je vhodné zmínit, že ve všech schématech uváděných v této kapitole je pro napěťovou větev 5 V použito označení „VCC“.



Obr. 6.5: Schéma zapojení 3,3V DC-DC měniče

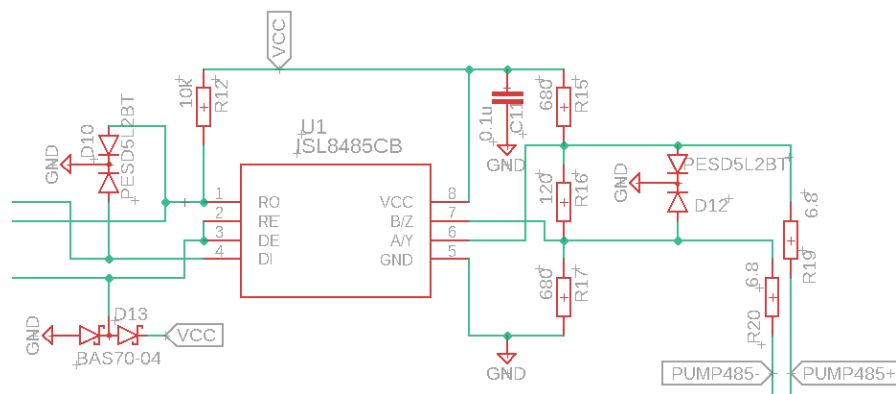
Místa, kde dochází ke spojení napájecích rozvodných cest shieldové desky a řídicí jednotky anebo vakuových periférií jsem vybavil ochranami proti přepětí, které by se šířilo jedním či druhým směrem. V případě výskytu takovéto napěťové špičky, sepnou výkonné ochranné prvky typu TVS (transient voltage suppressor), odvedou přebytečný proud a ochrání tak zařízení před poškozením. Příklad této ochrany

vidíme na Obr. 6.4, kde se kromě klasických usměrňovacích diod (D5, D6) nachází i inverzně orientovaná dioda D7. Jedná se právě o TVS, který omezuje vstupní napětí na přibližně 30 V. Je dobré zmínit, že se jedná o ochranu pro nestandardní případy, protože při běžném provozu bude celé zařízení napájeno ze zdroje 24 V a k sepnutí TVS tak nedojde. Podobné ochranné diody jsou umístěny i na rozvodech nižších napětí, ty už ale nelimitují napětí na 30 V, nýbrž přibližně 6 V.

Pro minimalizaci efektu vyšších napájecích proudů tekoucích signálovou cestou GND zpět do zdroje na citlivé signálové obvody, jsem po shieldové desce rozvedl dvojí zem. Konektory pro připojení periférií jsou umístěny v blízkosti konektoru napájecího a vzájemně jsou propojeny vždy dvojicí (+24 V a zem) silných tištěných spojů zajišťujících napájení. Ve schématech je tato „oddělená“ zem označena jako „AGND“. Volné prostory obou stran shieldové desky jsou vyplněny mědí, která spojuje zemnicí piny zbylých součástek a integrovaných obvodů. Oba zemnicí obvody se potom potkávají až v místě konektoru MOLEX 2 × 1.

### 6.3 Komunikace s turbomolekulární vývěvou

Pro zajištění komunikace s turbomolekulární vývěvou a jejím controllerem jsem z dostupných možností, které jsou výrobcem nabízeny [27], zvolil protokol standardního typu RS-485, který je v průmyslu hojně užíván a poskytuje nad pumpou největší kontrolu. Zprávami, které jsou tímto protokol mezi VSC a turbínou posílány, se zabývám v kapitole Sek. 7.1. Ústředním prvkem implementované komunikace RS-485 je integrovaný obvod ISL8485, který realizuje transformaci logických napěťových úrovní mezi UART, který je využíván řídicí jednotkou, a napěťovými úrovněmi užívanými protokolem RS-485. Integrovaný obvod jsem zapojil podle doporučení výrobce [31] a na vstup i výstup integrovaného obvodu jsem připojil diody chránící dané signálové cesty před přepětími.



Obr. 6.6: Schéma zapojení obvodu pro komunikaci s turbomolekulární vývěvou

Celé zapojení tohoto bloku je na Obr. 6.6. Vstupy integrovaného obvodu označené ve schématu jako RE a DI realizují přepínání mezi režimy vysílání a přijímání.

## 6.4 Komunikace s vakuovým ventilem

Zatímco turbomolekulární pumpa a její controller využívají standardní typ komunikačního protokolu, u ventilu se výrobce rozhodl sáhnout po atypickém řešení. Ke změně došlo i u posílaných zpráv, které jsou omezeny na jednoduché povely a dotazy – viz Sek. 7.2. Pro zprovoznění komunikace jsem potřeboval splnit kritéria daná produktovým listem použitého ventilu. [2] V této části textu se zabývám pouze hardwarovými implikacemi atypičnosti použitého způsobu sériové komunikace. Softwarové implikace (pomalá přenosová rychlost) rozebírám v Sek. 7.2.

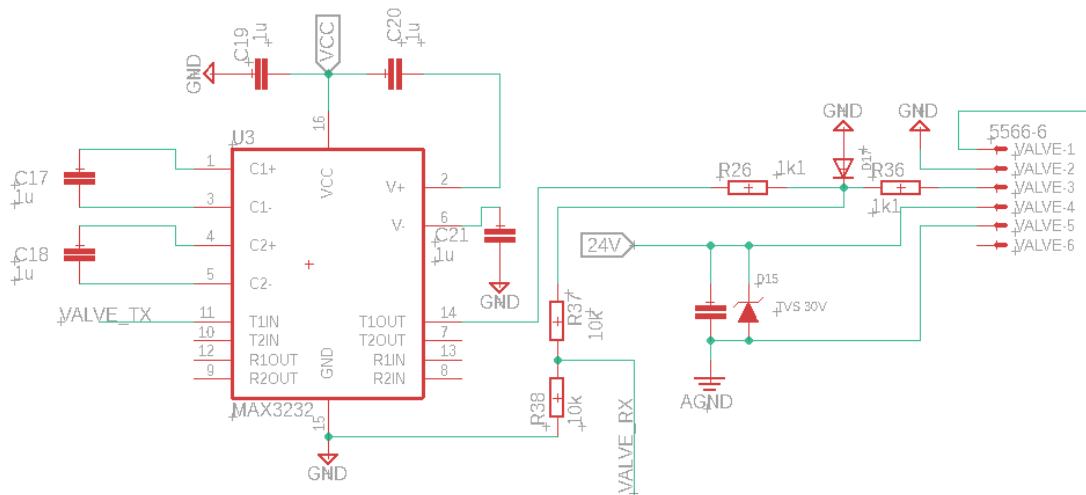
Podle podmínek z Obr. 6.7 jsem jako vhodné řešení zvolil podobný přístup jako v případě realizace komunikace s pumpou – tedy využít integrované řešení poskytující transformaci logických napěťových úrovní. Tentokrát však pro protokol RS-232, kde konkrétní zvolené řešení (integrovaný ob-

Data transmission		
Transmission rate		300 Baud
Data bits		7
Stop bits		2
Voltage level:	Logical 0	>7 V
	Logical 1	<3 V

Obr. 6.7: Požadavky na komunikaci s vakuovým ventilem [2]

vod MAX232) realizuje pro logické hodnoty binární výstup až  $\pm 15$  V pomocí nábojové pumpy. [34] Z informací dostupných od výrobce ventilu však není patrné, zda záporná napěťová úroveň na vstupu ventilu zařízení poškodí či nikoliv. Vzhledem k tomu, že se, stejně jako je tomu v případě i všech ostatních vakuových komponent, jedná o velmi drahý kus vybavení, přistoupil jsem k bezpečnému řešení a realizoval komunikační rozhraní se dvěma napěťovými úrovněmi:  $-0,7$  V a ca 8 V. Horní hranice je dána vstupním odporem ventilu a výkonem nábojové pumpy, naopak spodní hranice je dána úbytkem napětí na běžné usměrňovací diodě.

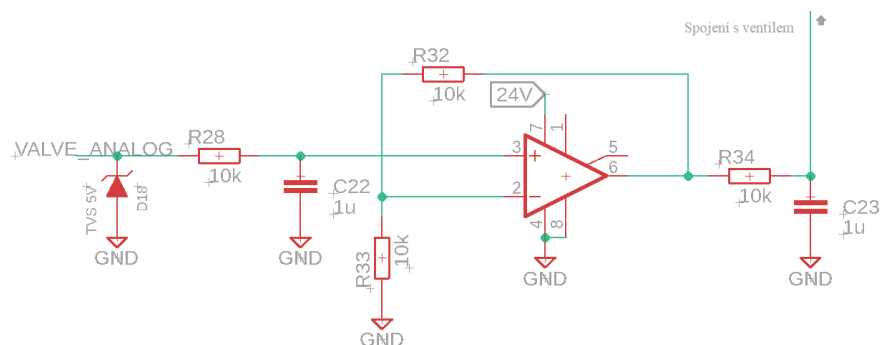
V případě, že je pomocí integrovaného obvodu MAX232 vysílána záporná napěťová úroveň, sepne se dioda na schématu Obr. 6.8 označená jako D17, protože na signální cestě bude nižší napětí, než je napětí GND (země). Napětí na vstupu ventilu tak bude rovné napětí 0 V umenšenému o napětí na diodě, které je přibližně rovno 0,7 V. Takto malé záporné napětí by již ventil poškodit nemělo a v praxi také nepoškodilo. Předřadný odpor R26 limituje proud diodou D17 a představuje tak určitý napěťový úbytek na komunikační cestě. Ten je však zanedbatelný a na správnou funkci komunikace mezi VSC a ventilem nemá vliv.



Obr. 6.8: Schéma zapojení obvodu pro komunikaci s vakuovým ventilem

Zpětná komunikace ze strany ventilu nabývá dvou napětových hodnot – 8 V a 2 V. Tyto úrovně nejsou vhodné pro zpracování pomocí MAX232, ale po podělení napětí pomocí vhodného rezistorového děliče (R37, R38) budou napětové úrovně poloviční – 4 V a 1 V – a vhodné pro přímé zpracování pomocí UART vstupů řídicí jednotky. Odpor R36 proudově chrání diodu při chybné funkci a výskytu záporného napětí ze strany ventilu.

Alternativním způsobem ovládní ventilu je využití jeho analogového vstupu, který na rozsahu 0 – 10 V ovlivňuje jeho pozici. [2] Pro rozšíření možností VSC jsem tento způsob ovládní na desku také implementoval. Využívá se při něm PWM (pulse width modulation) výstup řídicí jednotky, který je vhodným operačním zesilovačem zdvojnásoben na požadovaný rozsah do desíti voltů. Schéma tohoto zapojení je zobrazeno na Obr. 6.9.



Obr. 6.9: Schéma zapojení obvodu pro analogové ovládní vakuového ventilu

Výstupní pulzně modulovaný signál je nejprve vyhlazen pomocí RC článku, než je přiveden na vstup zesilovače v neinvertujícím zesilujícím zapojení. Zesílení je v



tomto případě dáno vztahem z Rce. (6.1).

$$K = 1 + \frac{R_{32}}{R_{33}} = 2 \quad (6.1)$$

Operační zesilovač tedy zesiluje přibližně dvojnásobně (zesílení se bude mírně lišit v závislosti na přesné hodnotě odporů), což je ideální, protože maximální výstupní napětí PWM pinu řídicí desky je 5 V, což je hodnota napájecího napětí desky. Pomocí tohoto pinu a PWM frekvence na něm je tak možné ovládat připojený ventil napětím 0–10 V. Pro správnou funkci je ale třeba zajistit, aby se operační zesilovač uměl přiblížit svému napájecímu napětí (alespoň na záporné napájecí větvi). Z produktového listu ventilu [2] vyplývá, že na rozsahu 0–10 V přibližně první volt ventil nereaguje na vstupní napětí změnou pozice. Operační zesilovač se tak nemusí těsně přiblížit spodní napájecí hranici 0 V, ale postačí, pokud bude schopný při přivedení 0 V na neinvertující vstup na výstupu dodat zhruba 1 V. Operační zesilovač nejběžnějšího typu LM741, tohoto pro vstupní odpor ventilu není schopen, jak dokazuje Obr. 6.10.

Output voltage swing	$V_S = \pm 15 \text{ V}$	$R_L \geq 10 \text{ k}\Omega$	$\pm 12$	$\pm 14$	V
		$R_L \geq 2 \text{ k}\Omega$	$\pm 10$	$\pm 13$	

Obr. 6.10: Výňatek z produktového listu OZ LM741 [35]

Pro tuto aplikaci jsem tak zvolil zesilovač s atributem *rail to rail*, který značí, že se výstupní napětí může velmi přiblížit (nebo i rovnat) napětí napájecímu. Konkrétně se jedná o model OPA189, který tyto nároky splňuje a je dobře dostupný. Odstup výstupního napětí od záporné napájecí větve je v případě tohoto zesilovače dostatečný, jak ukazuje Obr. 6.11, a možnosti pro přiblížení se kladné větvi jsou v této aplikaci irelevantní.

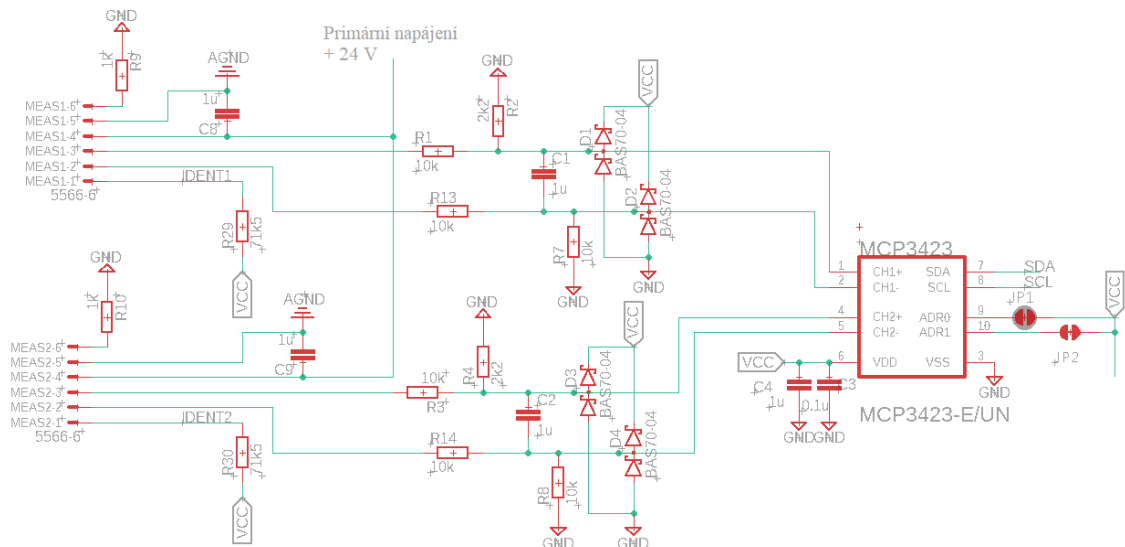
$V_o$	Voltage output swing from rail	Positive rail	No load	5	15	mV
			$R_{LOAD} = 10 \text{ k}\Omega$	20	110	
			$R_{LOAD} = 2 \text{ k}\Omega$	80	500	
		Negative rail	No load	5	15	
			$R_{LOAD} = 10 \text{ k}\Omega$	20	110	
			$R_{LOAD} = 2 \text{ k}\Omega$	80	500	
$T_A = -40^\circ\text{C to } 125^\circ\text{C, both rails, } R_{LOAD} = 10 \text{ k}\Omega$		OPA189 & OPA2189	20	120		
		OPA4189	20	140		

Obr. 6.11: Výňatek z produktového listu OZ OPA189 [36]

Na výstup operačního zesilovače jsem umístil ještě druhý RC článek, pro lepší vyhlazení napětí, kterým lze řídit ventil. Je také dobré podotknout, že tuto metodu ovládání ventilu jsem implementoval jako alternativní řešení, které ze své podstaty nabízí méně možností, než ovládání ventilu přes sériovou komunikaci. V dosavadní operaci VSC tak tuto cestu ovládání ventilu nebylo třeba využívat.

## 6.5 Vyčítání hodnot tlaku a identifikace měrek

Podporované vakuové měrky jako TPR-280 [28], PKR-251 [1] nebo libovolnou ze série PKR 36X [29] indikují jimi měřené tlaky pomocí analogového napětového diferenčního výstupu s rozsahem přibližně 0–10 V (liší se podle konkrétního typu). Byť má řídicí deska zabudované analogově-digitální převodníky, pro zpracování těchto hodnot jsem zvolil převodník externí, který nabízí více možností nastavení a především větší přesnost, viz Kap. 2. Jedná se o dvoukanálový převodník – zapojený podle schématu na Obr. 6.12 – který je ideální pro realizaci záměru umožnit vakuovou operaci s až dvěma vakuovými měrkami používanými současně.



Obr. 6.12: Schéma zapojení analogově-digitálního převodníku

Každá ze dvou vstupních větví je nejprve napětově upravena tak, aby maximální výstupní rozsah měřenosného napětí měrek odpovídal přibližně maximálnímu vstupnímu rozsahu MCP3423 a nepřekračoval jej. To zajišťuje vždy čtveřice rezistorů (R1, R2, R7 a R13, resp. R3, R4, R8 a R14), která zároveň slouží jako proudová ochrana dále umístěných diod. Kondenzátory C1 a C2 slouží pro vyhlazení průběhu signálu a diody D1, D2, D3 a D4 omezují maximální napětí, které se na vstupu převodníku může objevit, na +5 V a  $-0,7$  V. Součástky na pravé straně schématu z Obr. 6.12 označené jako JP1 a JP2 jsou cínem upravitelné propoje mezi vstupy převodníku ADR0 a ADR1 a napájecí pětivoltovou větví. Přivedením napětí, jejich uzemněním nebo ponecháním „ve vzduchu“ lze nastavovat adresu, pod kterou zařízení dále komunikuje po sběrnici I<sup>2</sup>C s řídicí jednotkou.

U každého z konektorů (označené jako MEAS1 a MEAS2) lze v Obr. 6.12 také nalézt signální větve označenou jako IDENT, na kterou je přes rezistor (R29 a R30)

přivedeno napájecí napětí 5 V. Tyto větve slouží pro identifikaci připojených měrek – každý typ podporované vakuové měrky má interně zapojený rezistor mezi pinem, na který je přiveden signál IDENT, a zemí o hodnotě odporu, který je unikátní pro daný model. Pro testované měrky TPR-280 a PKR-251 jsou potom hodnoty odporu uvedeny v Tab. 6.1

Tab. 6.1: Hodnoty identifikačních odporů testovaných měrek [28][1]

Model měrky	Typ použití	$R_{\text{ident}}$ [k $\Omega$ ]
TPR-280	-	3,0
PKR-251	Pouze Pirani	11,1
	Kombinovaně Cold Cathode a Pirani	9,1

Signální cesty IDENT jsou kromě pinů měrek připojeny i na piny řídicí desky, které jsou interně vybaveny analogově-digitálními převodníky s velkými vstupními odpory – lze proto napěťový dělič tvořený předřadným odporem  $R = 71,5 \text{ k}\Omega$  a identifikačním odporem považovat za nezatížený. Pomocí nich tak lze stanovit napětí na identifikačním odporu měrky  $U$  a z něj hodnotu identifikačního odporu  $R_{\text{ident}}$ :

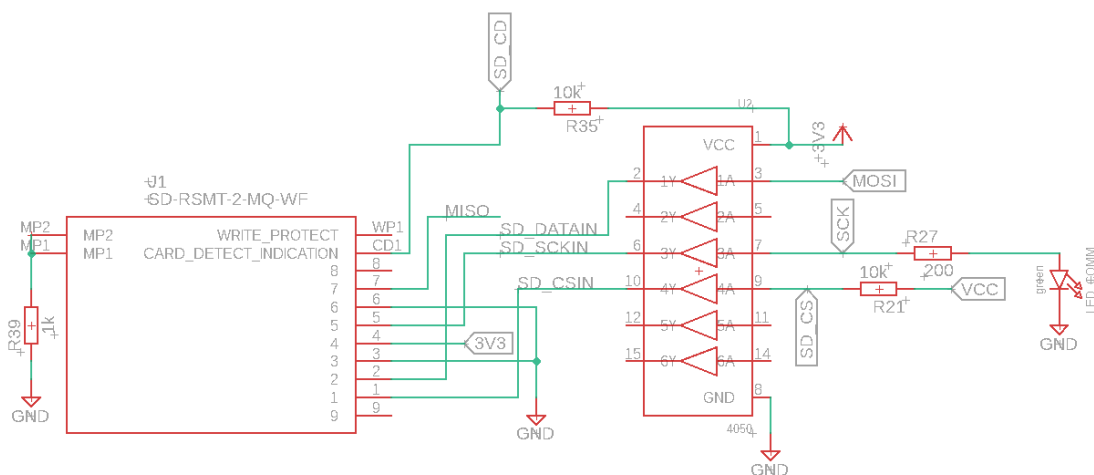
$$\frac{U}{U_{CC}} = \frac{R_{\text{ident}}}{R + R_{\text{ident}}} \quad (6.2)$$

$$R_{\text{ident}} = \frac{RU}{U_{CC} - U} \quad (6.3)$$

## 6.6 Slot SD karty

Pro vyhodnocování výsledků proběhlých experimentů je třeba zpracovat data při nich získaná. VSC umožňuje zápis uživatelem zvolených dat na SD kartu vloženou do patřičného slotu, pro který jsem na shieldové desce realizoval jednoduchý obvod.

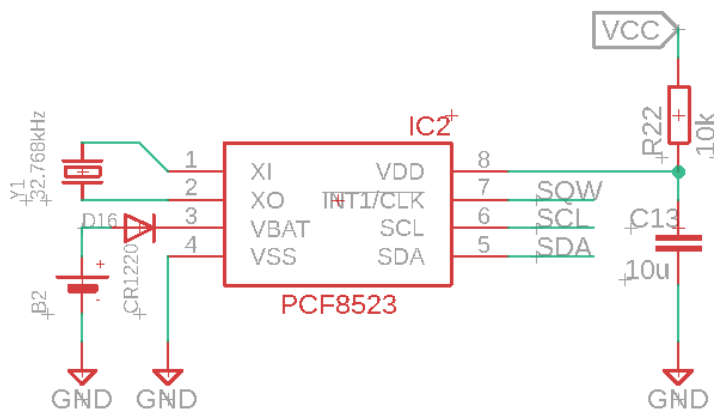
Zapojení je tvořeno samotným modulem pro SD kartu a integrovaným obvodem realizujícím funkci vyrovnávací paměti mezi řídicí jednotkou a SD kartou. Použitý slot je vybaven pinem, který po vložení paměťového média sepne k zemi, čehož je využito při vizualizaci stavu ukládání dat na displeji. Dioda na schématu Obr. 6.13 označená jako LED\_COMM slouží pro indikování probíhající komunikace.



Obr. 6.13: Schéma zapojení modulu pro připojení SD karty

## 6.7 Modul reálného času

Pro správný kontext ukládaných dat je třeba přiřadit k nim odpovídající časovou informaci. K uchování aktuálního času slouží modul reálného času RTC (real time clock), který si správný čas zachová i při odpojení napájení. K dodávání energie integrovanému obvodu PCF8523, který slouží pro výpočet aktuálního času, i v době, kdy není VSC připojeno k napájecímu zdroji, slouží baterie typu CR1220. Na schématu zapojení z Obr. 6.14 je potom patrný i krystalový oscilátor Y1, kterého integrovaný obvod využívá ke správnému odpočtu času. [25]



Obr. 6.14: Schéma zapojení modulu reálného času

Komunikace mezi integrovaným obvodem PCF8523 a řídicí jednotkou probíhá, podobně jako v případě analogově-digitálního převodníku (Sek. 6.5), pomocí sběrnice I<sup>2</sup>C. Modul RTC nemá nastavitelnou adresu na této sběrnici a ta je tak pevně dána jako 0x68. [25] Jelikož je ale druhé zařízení, které v rámci systému na této

sběrnici komunikuje (převodník) vybaveno přenastavitelnou adresou, nemůže dojít ke konfliktu a obě zařízení budou pomocí I<sup>2</sup>C správně obsloužena.

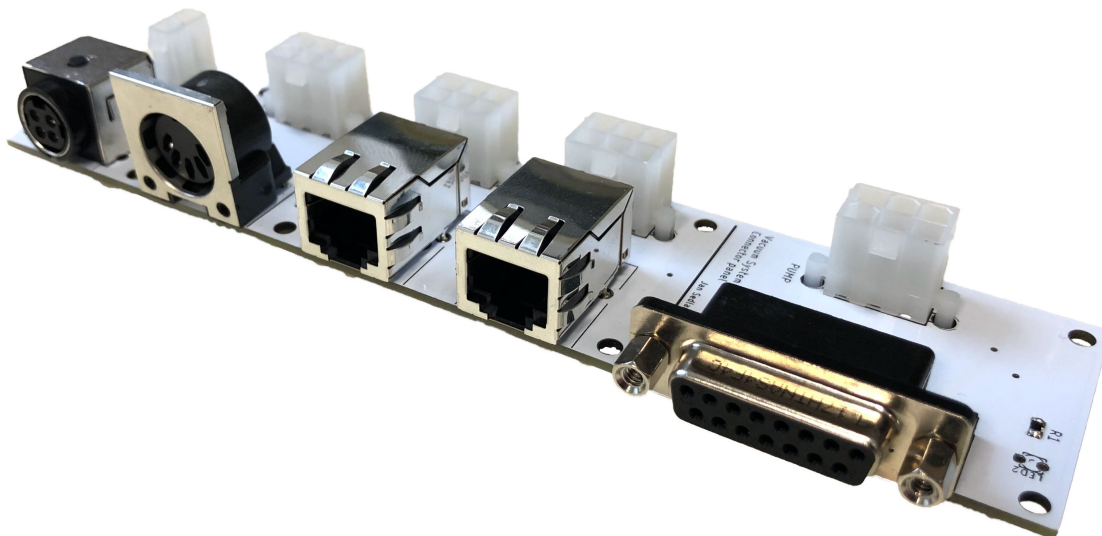
## 6.8 Konektory a fyzická propojení

Pro propojení mezi konektorovou deskou a vakuovými periferiemi jsem na míru vytvořil kabely, které jsou přizpůsobeny jak toku dostatečně velkých napájecích proudů, tak jsou i vhodné pro přenos komunikace mezi VSC a periferiemi. Všechny použité kabely jsou stíněné, aby byl omezen vliv rušení elektromagnetického pole generovaného zdrojem plazmatu. Pro každou periférii jsem na straně VSC zvolil takový konektor, který je relativně běžně užívaný, dostatečně proudově propustný a zamezuje špatnému připojení a poškození některého ze zařízení. Zkompletovaná deska osazená použitými konektory je zobrazena na Obr. 6.15.

K propojení měrek a VSC jsem zvolil stíněný ethernetový kabel s tloušťkou vodičů amerického standardu 26 AWG, který umožňuje proudový tok až přibližně 360 mA. [32] Vzhledem k výkonovému odběru měrek, který činí přibližně 2 W, je toto maximum více než dostatečné. Na straně kabelu, který slouží pro připojení k VSC, jsem využil typický konektor typu RJ-45, ale na straně pro připojení k měrce bylo třeba instalovat proprietární konektor výrobce *Pfeiffer Vacuum*.

Obdobný ethernetový kabel, ovšem o větší tloušťce vodičů 24 AWG, jsem využil pro propojení VSC a vakuového ventilu. Takto tlusté vodiče jsou již schopné přepravovat více než půl ampéru [32], což je pro ventil, jehož maximální proudový odběr činí 500 mA [2], dostatečné. Tohoto proudového odběru je navíc dosaženo, pouze pokud ventil mění svou pozici, a v klidových stavech nespotřebuje více než 30 mA proudu. Na jedné straně je tento kabel opět osazen proprietárním konektorem *Pfeiffer* a na straně pro připojení k VSC jsem zvolil kruhový konektor o vhodném počtu pinů.

Kabel pro připojení turbomolekulární vývěvy a jejího controlleru je ze tří výrobních kabelů jediný, který umožňuje oboustranné zapojení – na obou jeho koncích je osazen stejným konektorem se stejně vyvedenými piny. V případě controlleru TC110, přes který se k vývěvě samotné realizuje připojení, použil výrobce standardní konektor D-SUB o 15 pinech umístěných ve dvou řadách. Jelikož podporované pumpy mají výrazně větší spotřebu, než ostatní připojované periferie, byla třeba realizace alternativního postupu konstrukce kabelu. Problém vysokých napájecích proudů, které v případě použití nejvýkonnější z pump HiPace 80 činí až 4,6 A [4], jsem vyřešil rozdělením propojovacího kabelu na dva díly, z nichž jeden slouží čistě pro napájení pumpy a druhý zase pouze pro přenos komunikace. Oba kabely vychází z konektorového krytu, který je určen pro uskladnění dvou separátních kabelů. Komunikační kabel sestává z celkem 12 oddělených vodičů, ze kterých jsou aktuálně



Obr. 6.15: Umístění konektorů na konektorové desce

zapojené pouze dva sloužící pro přenos komunikace RS-485. Toto řešení však v budoucnu umožňuje plné propojení všech výstupů 15pinového konektoru, pokud by to mělo být žádoucí.

## 6.9 Napájení a propojení s displejem

Komunikaci se sedmipalcovým kapacitním displejem značky *Nextion* jsem po hardwarové stránce realizoval jednoduchým interním spojením řídicí jednotky Arduino Mega a displeje pomocí asynchronní komunikace UART, což je standardní a výrobcem doporučovaný způsob ovládání. [24] Napájení displeje zařizuje rozvodná pětivoltová větev, která zároveň napájí řídicí jednotku.

## 6.10 Kryt zařízení

Kryt zařízení je realizován jako plastový box, který uzavírá obě desky tištěných spojů osazené veškerou interní elektronikou a jejich vzájemné fyzické propojení. Vrchní strana zařízení, ve které je zapuštěn dotykový displej, je zkosená pod úhlem 30°, aby obsluze umožňovala pohodlnou práci se zařízením jak ve stoje, tak při sezení. Konektory pro připojení vakuových periférií jsou vyvedeny na zadní straně zařízení,

protože se předpokládá, že po zapojení přívodních kabelů je nebude třeba často odpojovat. Naopak slot pro umístění SD karty bude využíván při téměř každém provozovaném experimentu a je tak vyveden na boční stranu krytu. Aktuálně používanými materiály jsou plasty PLA a PETG černé barvy, které umožňují realizaci libovolného tvaru a rozměru krytu pomocí metody 3D tisku. Jako potenciální problém se může jevit jeho neschopnost chránit elektroniku před elektromagnetickými interferencemi, pro což by byl lepší kryt kovový. V praxi se však tato případná nevýhoda příliš neprojevila, jelikož zapojení vakuových měrek, jejichž napěťový měronosný signál by k těmto interferencím mohl být nejnáchylnější, je realizováno diferencně. Vzhled krytu VSC je zachycen na Obr. 6.16



Obr. 6.16: Zkompletovaný systém v krytu VSC

## 7 Softwarové zdroje

V této kapitole se věnuji popisu využití softwarových zdrojů v programu a popisují, jaké funkcionality jsem pro jejich další použití vytvořil. Celý projekt jsem napsal a zprovoznil v programovacím jazyku C++.

### 7.1 Pfeiffer Vacuum Protocol

Pro komplexnější zařízení implementuje výrobce vakuových komponent *Pfeiffer Vacuum* sadu speciálních zpráv, kterou označuje jako *Pfeiffer Vacuum Protocol*. V případě VSC lze tyto zprávy využít pro ovládání připojené turbomolekulární vývěvy a jejího controlleru. [27] Softwarovou implementaci tohoto protokolu jsem vytvořil jako knihovnu, jejíž objekt po vytvoření reprezentuje v kódu komunikaci s připojenou turbomolekulární vývěvou. Voláním metod tohoto objektu dochází k zasílání a čtení přijímaných zpráv. Implementovaná knihovna využívá hardwarové komunikace RS-485, jejíž realizaci jsem popsal v Sek. 6.3.

Nejdůležitějšími metodami knihovny *PVProtocol* jsou tyto tři, které realizují zasílání a přijímání zpráv: `requestData`, `controlCommand` a `readResponse`. Metoda `requestData` zasílá požadavek na informaci o některém z údajů týkajících se pumpy nebo jejího aktuálního provozu, `controlCommand` ovládá pumpu a přepisuje tak některé její parametry (jako například aktuální rychlost otáčení) a `readResponse` zajišťuje správné zachycení a přečtení přijímané zprávy. Než uvedu podobu jejich konkrétní implementace, je třeba se podívat na předepsaný vzhled zpráv *Pfeiffer Vacuum Protocolu* na Obr. 7.1.

a2	a1	a0	*	0	n2	n1	n0	l1	l0	dn	...	d0	c2	c1	c0	C <sub>R</sub>
a2 – a0	Unit address for slave ○ <ul style="list-style-type: none"><li>• Individual address of the unit ["001";"255"]</li><li>• Group address "9xx" for all identical units (no response)</li><li>• global address "000" for all units on the bus (no response)</li></ul>															
*	Action according to telegram description															
n2 – n0	Pfeiffer Vacuum parameter numbers															
l1 – l0	Data length dn to d0															
dn – d0	Data in the respective data type (see chapter "Data types used", page 22).															
c2 – c0	Checksum (sum of ASCII values of cells a2 to d0) modulo 256															
C <sub>R</sub>	carriage return (ASCII 13)															

Obr. 7.1: Formát zpráv *Pfeiffer Vacuum Protocolu* [27]

V každé zprávě se nejprve zasílají tři byty adresy zařízení, kterému je daná zpráva na sběrnici určena, poté číslo parametru, který je měněn nebo od pumpy



vyžádán, dále případně hodnota, která má být do stanoveného parametru zapsána, kontrolní suma a znak *carriage return* s ASCII kódem 13 pro ukončení zprávy. Pro ilustraci podoby implementace uvedu kód jedné metody, která slouží pro zaslání požadavku pumpě (`controlCommand`), a druhou pro přijetí a zpracování zprávy (`readResponse`).

Metoda `controlCommand` s kódem uvedeným ve Výp. 7.1 probíhá v následujících krocích: nejprve nastaví pro integrovaný obvod ISL8485 režim vysílání přivedením logické jedničky na jeho vstupy RE a DI, viz schéma na Obr. 6.6. Následně je podle Obr. 7.1 vytvořen řetězec znaků, které tvoří tělo zprávy obsahující adresu pumpy, číslo parametru a hodnotu, která mu má být přiřazena. Na konec této zprávy je připojen kontrolní součet a ASCII znak číslo 13. Pokud je v UART bufferu dostatek volných bytů pro odeslání této zprávy, učiní se tak, a ISL8485 se okamžitě přepne do režimu přijímání zpráv.

#### Výpis 7.1: PVP: metoda `controlCommand`

```
1 void PVPProtocol::controlCommand(const char* address, const char* parameter
  ↪ , String dataString) {
2   digitalWriteFast(RS485PIN, HIGH);
3   int sum = 0;
4   time_last = millis();
5   String serialString = address;
6   serialString += "10";
7   serialString += parameter;
8   if(dataString.length()<9) serialString += "0";
9   serialString += String(dataString.length());
10  serialString += dataString;
11  for (char i : serialString) {
12    sum += i;
13  }
14  if (sum%256 < 10) serialString += "00";
15  else if (sum%256 < 100) serialString += "0";
16  serialString += String(sum%256);
17  serialString += '\r';
18  while (PfeifferSerial->availableForWrite() < serialString.length())
19  {
20    if ((millis() - time_last) > 400) return;
21  }
22  PfeifferSerial->print(serialString);
23  PfeifferSerial->flush();
24  digitalWriteFast(RS485PIN, LOW);
25 }
```

Po odeslání zprávy nadchází čas přijmout zprávu, kterou turbomolekulární vývěva zašle jako reakci zpět. To realizuje zmiňovaná knihovní metoda `readResponse` implementovaná podle Výp. 7.2. Ta nejprve ověří nastavení správného režimu ISL8485 pro příjem dat a poté zkontroluje, že přišel minimálně takový počet bytů, které by mohla obshovat nejkratší možná přijatá zpráva. Pokud se tak stalo, metoda čte byty z bufferu, dokud nenarazí na znak určující konec zprávy. Dále probíhá kontrola správného formátu zprávy – korektní ukončení, správná délka a správný kontrolní součet. Zpráva je následně rozparsována na její jednotlivé části, které jsou pro případné další zpracování uloženy do prvků struktury `pumpResponse`. Pro návratovou hodnotu samotné metody se však využije pouze nosná informace zprávy, v tomto případě tedy nastavená hodnota požadovaného parametru. V případě, že vše proběhlo správně, se tato hodnota rovná hodnotě, která byla v pumpě zaslána v povelu pomocí metody `controlCommand`. Jestliže byl nejprve zaslán požadavek metodou `requestData`, nosná informace obsahuje vyžádaná data.

Výpis 7.2: PVP: `readResponse`

```
1 String PVProtocol::readResponse() {
2     digitalWriteFast(RS485PIN, LOW);
3     String serialString;
4     String dataString;
5     int serialChar;
6     time_last = millis();
7     int sum = 0;
8     int checksum = 0;
9     while (PfeifferSerial->available() < 14) {
10         if (millis() > time_last + 400) return "RTTL";
11     }
12     for (int i = 0; i < 2500; ++i) {
13         serialChar = PfeifferSerial -> read();
14         if (serialChar == -1) continue;
15         serialString += serialChar;
16         if (int(serialChar) == 13) break;
17     }
18     if (serialString.length() < 14) return "Invalid_gauge_response!";
19
20     if (int (serialString [serialString.length() - 1]) != 13)
21         return "gauge_response_invalidly_terminated";
22
23     for (unsigned l = 0; l < serialString.length() - 4; ++l)
24         sum += serialString[l];
25     for (unsigned l = serialString.length()-4; l < 3; ++l)
```

```

26     checksum += serialString[1];
27     checksum = 100 * (serialString[serialString.length()-4] - 48);
28     checksum += 10 * (serialString[serialString.length()-3] - 48);
29     checksum += (serialString[serialString.length()-2] - 48);
30     if (checksum != sum % 256) return "Invalid_checksum!";
31
32     dataString = "";
33     for (int l = 0; l < 3; ++l) dataString += serialString[l];
34     pumpResponse.address = dataString;
35     dataString = "";
36     for (int l = 3; l < 5; ++l) dataString += serialString[l];
37     pumpResponse.rw = dataString;
38     dataString = "";
39     for (int l = 5; l < 9; ++l) dataString += serialString[l];
40     pumpResponse.parameter = dataString;
41     dataString = "";
42     for (unsigned l = 10; l < serialString.length() - 4; ++l)
43         dataString += serialString[l];
44     pumpResponse.data = dataString;
45
46     return dataString;
47 }

```

## 7.2 Ovládání vakuového ventilu

Jak jsem již zmínil v Sek. 6.4, pro ovládání ventilu nebylo možné využít protokol a sériovou komunikaci standardnějšího provedení, jako tomu je u turbomolekulární vývěvy. Specifika výrobcem implementované komunikace se při softwarové implementaci projevují především nízkou přenosovou rychlostí 300 baudů, viz Obr. 6.7, pro kterou nelze využít klasické hardwarové UART na řídicí jednotce. Také u komunikace s vakuovým ventilem platí, že logická hodnota 0 nabývá na lince vyšší napěťové hodnoty, než logická 1 – jedná se tedy o komunikaci s inverzní logikou. Pro realizaci sériové komunikace s takto pomalou rychlostí a inverzní logikou jsem zvolil komunitní knihovnu `SoftwareSerial`<sup>1</sup>. Ta umožňuje provozovat sériovou komunikaci i na jiných pinech řídicí jednotky Arduino Mega, než na těch, které jsou k tomuto účelu určeny z výroby. Neposkytuje potom stejně dobré vlastnosti po stránce výkonu, ale pro tuto aplikaci je knihovna postačující.

<sup>1</sup>Creative Commons Attribution Share Alike 4.0 International, bez vlastních úprav, <https://docs.arduino.cc/learn/built-in-libraries/software-serial>

Konkrétní vzhled zpráv, které je třeba ventilu zasílat, uvádí jeho produktový list. Na Obr. 7.2 je vidět, že zpráva je zahájena malým písmenem „g“ až „z“, které označuje parametr, jež má zpráva nastavit. Za písmenem následuje několik hexadecimálních znaků, jejichž počet se liší parametrem od parametru. Udávají hodnotu, na kterou se má zvolený parametr nastavit. Zpráva je potom zakončena dvojicí ASCII znaků s kódy 13 a 10. V případě, že ventil zprávu úspěšně přijal a zpracoval, zašle stejnou zprávu zpět, pouze s velkým počátečním písmenem.

#### Communication

Each transmission from the controller to the valve is initiated with one ASCII character from "g" to "z" (67<sub>h</sub> to 7A<sub>h</sub>) and terminated with "CR/LF" (0D<sub>h</sub>, 0A<sub>h</sub>).

Numeric transmission data are represented as HEX 2 or 3 position hexadecimal values.

For transmission to the valve 0 ... 9 and a ... f are used, for transmission to the controller 0 ... 9 and A ... F.

For two digit numbers a +/- sign can additionally be specified.

g\$\$\$	G\$\$\$	Go to absolute position \$\$\$ x2
		Examples:
		g100 (close)⇒ Response G100 (= Absolute position 0200 <sub>h</sub> )
		gd34 (open)⇒ Response GD34 (= Absolute position 1A68 <sub>h</sub> )

(a) Popis formátu zpráv

(b) Příklad zprávy

Obr. 7.2: Produktový list popisující zprávy pro ovládání ventilu [2]

Jsou také definovány jednoduché zprávy, které nenesou žádnou další hodnotu a pouze říkají, aby se ventil plně otevřel nebo uzavřel. Posledním typem zpráv jsou takové, které od ventilu vyžadují nějakou informaci. Tyto zprávy se sestávají s písmene a otazníku, který značí vyžádání dat.

Pro ovládání komunikace s ventilem jsem podobně jako v případě komunikace s turbomolekulární vývěvou na míru vytvořil knihovnu, tentokrát s názvem `ValveProtocol`, jejíž metody komunikaci obsluhují. Z důvodu popisovaného v následujícím odstavci jsem nejprve musel vytvořit jednoduché metody pro odesílání a přijímání

Inquiry commands		
Command	Response	Description
h?	H\$\$	Output the VMODE
p?	\$\$\$\$	Actual position (Normal range 0200H to 1A68H)
s?	S\$\$\$	Status information (12 Bit)
t?	T\$\$\$	Temperature in valve (12 Bit)
v?	V\$\$\$	Version number (=V120)

Obr. 7.3: Vyžádání informací od ventilu [2]

samostatných bytů a posléze teprve metody, které odesílají nebo zpracovávají celé řetězce znaků. Vzhledem k malému množství možných zpráv, jsem přímo vytvořil metody pro posouvání ventilu, jeho uzavírání nebo zjištění teplot, a nikoliv pro vytvoření zprávy obecného formátu, tak jako tomu bylo u vývěvy (Sek. 7.1). Knihovna `SoftwareSerial` může mít několik aktivních instancí (objektů) a ke každému z nich

jsou přiřazeny Rx (přijímací) a Tx (vysílací) piny, ale pouze jeden z nich může aktivně naslouchat. Vzhledem k hardwarové realizaci komunikace mezi řídicí deskou a ventilem (Sek. 6.4), jsem pro správnou funkci knihovny musel vytvořit dva objekty – jeden určený pro vysílání zpráv (za pomoci MAX232) s inverzní logikou a druhý pro přijímání zpráv (přes odporový dělič) s logikou klasickou.

Jak je ukázáno na Obr. 6.7, data je třeba přenášet formátem 7N2 – sedm datových bitů a 2 bity ukončovací, paritní bit není využit. Vzhledem k tomu, že standardní přenos probíhá formátem 8N1, provedl jsem při odesílání a zpracování každého bytu jednoduchou úpravu: pro odesílání dat je nejprve MSB (most significant bit), který reprezentuje první ze dvou stop bitů, nastaven na hodnotu 1, a naopak při přijímání dat je MSB nastaven na nulu, aby jeho hodnota nehrála roli. Obě jednoduché metody jsou uvedeny ve Výp. 7.3. Je dobré si také povšimnout, že každá z metod pracuje s jiným objektem (`valveSerial` a `valveSerialReceive`), jak jsem již zmínil dříve.

Výpis 7.3: ValveProtocol: Metody pro čtení a zápis jednoho bytu

```
1 char ValveProtocol::read() {
2     return char(valveSerialReceive->read() & 0x7F);
3 }
4
5 size_t ValveProtocol::write(uint8_t a) {
6     return valveSerial->write(a | 0x80);
7 }
```

Na těchto jednoduchých metodách jsem dále vystavěl metody pro přijímání a zasílání celých zpráv (Výp. 7.4) a dále pak samotné ovládání ventilu. Metoda `sendCommand` pouze ke zprávě předané argumentem přidá dva znaky (ASCII 13 a 10), které značí konec zprávy, a po jednotlivých bytech s využitím metody `write` (Výp. 7.3) zprávu odešle. Metoda `readResponse` se nejprve ujistí, že objekt knihovny `SoftwareSerial` určený pro přijímání zpráv je nastaven pro aktivní poslouchání a dále čeká, než je zachycen požadovaný počet bytů, které jsou metodě předány jejím parametrem. Dva bity navíc jsou opět dva znaky (ASCII 13 a 10) značící konec zprávy. Pokud byla zachycena zpráva o požadované délce, metoda jednotlivé byty přečte a pokud jsou platné, uloží je do výstupního řetězce.

Výpis 7.4: ValveProtocol: Metody pro odesílání a přijímání zpráv

```
1 void ValveProtocol::sendCommand(String command) {
2     command += "\r\n";
3     for (unsigned i = 0; i < command.length(); ++i)
4         write(command.charAt(i));
5 }
6
```

```

7 String ValveProtocol::readResponse(uint8_t length) {
8     valveSerialReceive->listen();
9     unsigned long t = millis();
10    while (valveSerialReceive->available() < length + 2) {
11        if (millis() - t > 400) return "TIMEOUT";
12    }
13    char response[length + 1];
14    size_t i = 0;
15    while(valveSerialReceive->available()) {
16        auto k = read();
17        if (!k) continue;
18        if (k == '\n' || k == '\r') continue;
19        response[i++] = k;
20    }
21    response[length] = 0;
22    return {response};
23 }

```

Pro příklad ještě ve Výp. 7.5 uvedu metodu pro nastavení konkrétní pozice ventilu, která pro zaslání příkazu využívá `sendCommand` a pro ověření jeho přijetí `readResponse` (Výp. 7.4). Metoda `setPosition` nejprve zkontroluje, že požadovaná pozice je v povoleném rozsahu hodnot. Do objektu označeného „s“, který slouží k ukládání stavů programu, uloží požadovanou hodnotu pozice ventilu předanou metodě pomocí parametru. Dále vytvoří zprávu pro odeslání a tu pomocí `sendCommand` odešle. Po přijmutí odpovědi ještě zkontroluje, že byl požadavek ventilem zpracován – první písmeno vrácené zprávy je „G“.

Výpis 7.5: ValveProtocol: nastavení ventilu do konkrétní pozice

```

1 bool ValveControl::setPosition(uint16_t pos) {
2     String cmd = "g";
3     if (pos > 6760) pos = 6760;
4     if (pos < 512) pos = 512;
5     s.valvePositionTarget = pos;
6     cmd += String(int(pos/2), HEX);
7     valve->sendCommand(cmd.c_str());
8     return valve->readResponse(4)[0] == 'G';
9 }

```

V kapitole Sek. 6.4 jsem se věnoval i alternativnímu způsobu ovládání ventilu přes jeho napěťový vstup. Proto jsem implementoval obvod realizující napěťový výstup VSC o rozsahu 0–10 V, který lze za tímto účelem použít. Aktuálně však není potřeba tento způsob ovládání využívat a kromě jednoduchého nastavení PWM frekvence na příslušném pinu jsem se tak jeho softwarovou implementací dále nezabýval.

## 7.3 Obsluha vakuových měrek

Pro softwarovou obsluhu k VSC připojených vakuových měrek je nutná realizace dvou funkcí: rutinní vyčítání hodnoty tlaku a jednorázová identifikace typu připojené měrky, která určuje převodní konstanty měřeného napětí na hodnotu tlaku.

### 7.3.1 Identifikace připojených měrek

Identifikace připojených měrek spočívá v určení napětí na pinech řídicí jednotky, které jsou připojeny na signálové cesty MEAS ve schématu na Obr. 6.12. Maximální napětí, které jsou piny schopné při aktuálním nastavení zpracovat, je 1,1 V. Napětí, ze kterého je odporový dělič sloužící pro identifikaci napájen (popisoval jsem jej v Sek. 6.5), má hodnotu 5 V. Maximální hodnota identifikačního odporu, který je tímto způsobem možné zpracovat, je proto po dosažení do Sek. 6.5  $R \approx 20166 \Omega$ .

Samotné určení typu k VSC připojené měrky zařizuje metoda `ident` z knihovny `gauge`, která obecně zajišťuje obsluhu měrek a operaci s nimi.

Výpis 7.6: `gauge`: Identifikace připojených měrek

```
1 void gauge::ident() {
2     analogReference(INTERNAL1V1);
3     uint8_t pin = A8;
4     if (this->channel == 1) pin = A9;
5     int voltageAn = analogRead(pin);
6     double voltage = 1.1 * voltageAn / 1023;
7     double current = (5 - voltage) / 71500;
8     double R = voltage/current;
9     gaugeTypeE* target = nullptr;
10    if (this->channel == 0) target = &s.gauge1;
11    if (this->channel == 1) target = &s.gauge2;
12    for (auto i : resList) {
13        if (eqApprox(R, i.res)) {
14            *target = i.gauge;
15            return;
16        }
17    }
18    *target = gaugeNotConnected;
19 }
```

Metoda nejprve nastaví správnou interní referenci (na zmiňovaných 1,1 V) a podle toho, k identifikaci měrky na jakém kanálu dochází, nastaví příslušný pin. Dále je podle Sek. 6.5 spočten identifikační odpor zapojen uvnitř měrky. Následně je

procházeno pole struktur `resList`, které ukládá výrobcem udávané hodnoty identifikačních odporů a modelů měrek jim příslušících. Funkce `eqApprox` porovná s přesností na 7% hodnotu odporu udanou výrobcem a hodnotu změřenou. Pokud se porovnávané hodnoty rovnají, byl úspěšně identifikován model připojené měrky.

Tuto metodu využívám také pro rozpoznání, zda-li je nějaká měrka vůbec připojena. Pokud nebude v poli `resList` nalezena žádná shoda, příslušný kanál je označen atributem, který říká, že k němu není připojena žádná měrka.

### 7.3.2 Stanovení hodnot měřeného tlaku

Jak jsem uváděl v Sek. 8.2.1, měronosným signálem z vakuových měrek jsou napěťová hodnota na vstupu analogově-digitálního převodníku. Pro zvolený integrovaný A/D převodník MCP3423 jsem se rozhodl využít veřejně dostupnou knihovnu MCP342x<sup>2</sup>. Ta obsahuje obslužné metody převodníku a uživatelé poskytuje jednoduchou kontrolu nad voláním převodu. Moje implementace této knihovny pro získání digitální hodnoty odpovídající měronosnému napětí je uvedena ve Výp. 7.7. V ní obsažená metoda `pressureMeas` provede jednoduché vyčtení digitální hodnoty z příslušného kanálu převodníku MCP3423 a provede přepočtení této hodnoty na napětí. Ke každému typu podporované měrky je kromě jejího názvu a hodnoty identifikačního odporu v programu přiřazena i převodní funkce, díky které je možné z naměřené napěťové hodnoty stanovit tlak v aparatuře. Tento převod realizuje funkce `voltageToPressure` podle převodních vztahů daných výrobcem v produktovém listu. Tyto vztahy jsou uvedeny v Obr. 7.5.

$p_{eff} = K \times \text{pressure indicated}$		
where	gas type	K
	air (N <sub>2</sub> , O <sub>2</sub> , CO)	1.0
	Xe	0.4
	Kr	0.5
	Ar	0.8
	H <sub>2</sub>	2.4
	Ne	4.1
	He	5.9

These conversion factors are average values.

Obr. 7.4: Vztah pro přepočtení skutečné hodnoty tlaku podle plynu přítomného v aparatuře [1]

<sup>2</sup>GNU Lesser General Public License v2.1, stevemarle, bez úprav, <https://github.com/stevemarle/MCP342x>



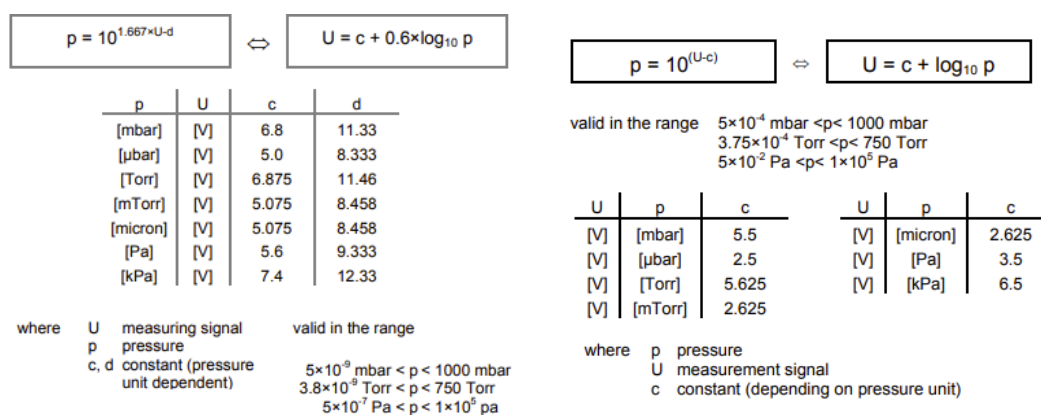
Výpis 7.7: gauge: stanovení tlaku z napěťové hodnoty

```

1 uint8_t gauge::pressureMeas(double &value) {
2     MCP342x::Config status;
3     uint8_t err;
4     long valueADC = 0;
5     double voltage = 0.0;
6     MCP342x::Channel ch = this->channel == 0 ? MCP342x::channel1 : MCP342x
    ↪ ::channel2;
7     err = adc.convertAndRead(ch, MCP342x::oneShot, MCP342x::resolution18,
    ↪ MCP342x::gain1, 200, valueADC, status);
8     voltage = 0.0000873314 * double(valueADC) - 0.0004127629;
9     value = voltageToPressure(this->type, voltage);
10    return err;
11 }

```

Grafické rozhraní displeje Nextion také umožňuje uživateli při operaci zvolit typ plynu, který v aparatuře převládá. Správná hodnota tlaku je potom stanovena z hodnoty získané pomocí převodních vztahů, která je vynásobena příslušným koeficientem pro daný plyn Obr. 7.4.



(a) PKR-251 [1]

(b) TPR-280 [5]

Obr. 7.5: Převodní vztahy  $U \rightarrow p$  pro vakuové měřky

## 7.4 Ukládání dat na SD kartu

V Sek. 6.6 jsem ukázal, že na shieldové desce zařízení je instalován slot pro SD kartu. Skrz otvor v krytu VSC je do něj možné tuto kartu vložit a ta posléze slouží pro ukládání uživatelem zvolených dat. Softwarově jsem tuto funkcionalitu realizoval

pomocí předinstalované knihovny pro Arduino SD.h.<sup>3</sup> V Sek. 6.6 zmiňovaný pin sloužící pro stanovení vložení karty je v kódu využit tak, aby uživatele na tuto skutečnost upozornil a aby nedošlo ke zbytečné ztrátě naměřených dat.

## 7.5 Reálný čas

Pro softwarové zabezpečení modulu reálného času jsem opět sáhnul po předpřipravené knihovně pro Arduino RTCLib<sup>4</sup>, která mimo jiné umožňuje i práce s použitým integrovaným obvodem PCF8523 (Sek. 6.7). Správný čas je do modulu nahrán při každém startu zařízení, pokud je připojené k počítači. Aktuální přesný čas je v zařízení využíván v podobě textového řetězce, který je buď zobrazován na displej, nebo je přikládán do souboru k ukládaným datům měření. Tento řetězec je získán zavoláním jednoduché funkce `rtcString` uvedené ve Výp. 7.8. V funkci je nejprve vytvořen objekt s aktuálním časem, ke kterému je přičteno 17 sekund. To je zhruba doba, která proběhne od nahrání kódu (při kterém je čas aktualizován) do okamžiku, než je zařízení spuštěno a modul a časové informace předány modulu reálného času. Dále jsou z objektu vybírány informace o dnech, hodinách apod. a standardním způsobem formátovány do textového řetězce pro další použití.

Výpis 7.8: Tvorba textového řetězce aktuálního času

```
1 String rtcString () {
2     rtclk = rtc.now() + TimeSpan(0,0,0,17);
3     String clkDataString = String(rtclk.day()) + "." + String(rtclk.month
4         ↪ ()) + "." + String(rtclk.year()) + "_-";
5     if (rtclk.hour() < 10) clkDataString += "0";
6     clkDataString += String(rtclk.hour()) + ":";
7     if (rtclk.minute() < 10) clkDataString += "0";
8     clkDataString += String(rtclk.minute()) + ":";
9     if (rtclk.second() < 10) clkDataString += "0";
10    clkDataString += String(rtclk.second());
11    return clkDataString;
}
```

## 7.6 Komunikace s displejem

Konkrétní podoba příkazů zasílaných řídicí jednotkou displeji vychází z online dostupné dokumentace výrobce. [24] Jedná se o celou sadu různých zpráv, které umož-

<sup>3</sup><https://github.com/arduino-libraries/SD>

<sup>4</sup>MIT License, Autor: Jeelab, bez úprav, <https://github.com/adafruit/RTCLib>

ňují nastavování vlastností jednotlivých prvků zobrazovaných na displeji, obsluhu integrovaného slotu pro micro-SD kartu i zasílání uživatelsky definovaných sekvencí znaků, které slouží pro spouštění specifických funkcí na straně řídicí jednotky, pokud dojde ke stisku konkrétního tlačítka na displeji.

Pro zabezpečení komunikace jsem zvolil hotovou knihovnu `EasyNextion`<sup>5</sup>, jelikož knihovna `ITEADlib` [24] poskytovaná výrobcem displeje se neukázala být v praxi spolehlivou. `EasyNextion` nabízí několik základních příkazů umožňujících zasílání textových (`writeStr`) nebo číselných příkazů (`writeNum`) a také čtení textových (`readStr`) i číselných (`readNumber`) hodnot parametrů prvků na displeji. Pomocí vzájemné kombinace těchto příkazů jsou obsluhovány veškeré funkcionality, které vyžadují čtení údajů z displeje nebo jejich změnu. Pro ilustraci jejich použití uvádím ve Výp. 7.9 funkci `peripheralsConnectCheck`, která po zavolání zjistí, zda-li jsou k VSC připojené periferie a pokud ano, určí konkrétní model těchto periférií (pro ty, které to umožňují).

Výpis 7.9: Kontrola připojených periférií

```
1 void peripheralsConnectCheck() {
2     gauge1.ident();
3     gauge2.ident();
4
5
6     if (pump.requestStatus("399") == "RTTL") s.pumpType = pumpNotConnected
       ↪ ;
7     else s.pumpType = PumpControl::stringToType(nex.readStr("main.pump.txt
       ↪ "));
8     if (valve.getPosition() < 30) s.valveType = valveNotConnected;
9     else s.valveType = evr116;
10
11     unsigned pic = pump.pumpPic(), picL = pic + 1;
12     nex.writeNum("peripherals.pumpPic.pic", picL);
13     nex.writeStr("main.pump.txt", pump.pumpName());
14     nex.writeStr("peripherals.pump.txt=main.pump.txt");
15
16     pic = valve.valvePic(), picL = pic + 1;
17     nex.writeStr("main.valve.txt", valve.valveName());
18     nex.writeStr("peripherals.valve.txt=main.valve.txt");
19     nex.writeNum("peripherals.valvePic.pic", picL);
20 }
```

---

<sup>5</sup>MIT License, Autor: Seithan, bez úprav, <https://github.com/Seithan/EasyNextionLibrary>

```

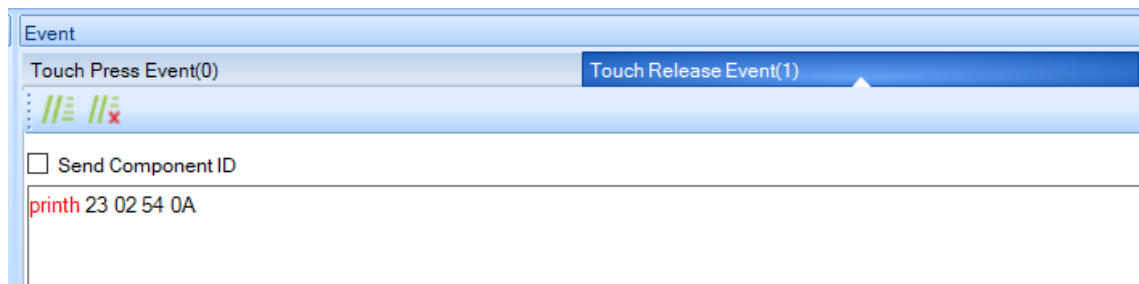
21  pic = gauge1.gaugePic(), picL = pic + 1;
22  nex.writeStr("main.gauge1.txt", gauge1.gaugeName());
23  nex.writeStr("meas.gauge1.txt=main.gauge1.txt");
24  nex.writeNum("meas.gauge1pic.pic", pic);
25
26  ...
27
28  if (s.gauge1 == gaugeNotConnected) {
29      nex.writeStr("meas.conn1.txt", "Not_connected");
30      nex.writeNum("meas.conn1.pco", 49152);
31  }
32  else {
33      nex.writeStr("meas.conn1.txt", "Connected");
34      nex.writeNum("meas.conn1.pco", 1472);
35  }
36
37  ...
38
39  if (s.gauge1 == gaugeNotConnected || s.gauge1 == gaugeUnused) s.
    ↪ CH1active = false;
40  if (s.gauge2 == gaugeNotConnected || s.gauge2 == gaugeUnused) s.
    ↪ CH2active = false;
41 }

```

Akce po zavolání funkce `peripheralsConnectCheck` jsou následující: nejprve proběhne identifikace připojených měrek pomocí knihovní metody `gauge::ident` (viz Výp. 7.6) a poté se stanoví, zda-li je připojena turbomolekulární vývěva a ventil. Podle stanoveného typu připojených měrek a uživatelem určeného typu pumpy a ventilu se na displej pomocí příkazů `writeStr` volaných na objekt displeje `nex` zapíše jejich názvy a příkazem `writeNum` se zase do prvků představujících obrázky na displeji zapíše identifikátory příslušných obrázků periferií. Poté je obdobným způsobem do textových polí displeje zapsán stav připojení jednotlivých periferií a nakonec je stanoveno, jestli mají být kanály analogově-digitálního převodníku využívány při provádění měření tlaku.

Pro spouštění specifických akcí po stisku tlačítka na displeji nabízí knihovna `EasyNextion` sadu funkcí nazvaných jako `trigger`. Ty spolupracují s displejem, jehož příslušná tlačítka jsou při tvorbě uživatelského rozhraní nastavena tak, aby po jejich stisknutí došlo k vyslání řady specifických znaků, jak ukazuje Obr. 7.6, po sériové lince k řídicí jednotce.

Tyto zprávy mají vždy pevně dané znaky „23 02 54“ za nimiž následuje dvojice bytů hexadecimálně reprezentující číslo `trigger` funkce, která má být na straně



Obr. 7.6: Příklad zaslání specifických znaků pro spuštění `trigger` funkcí po stisku tlačítka na displeji

řídící jednotky zavolána. Pro příklad takovéto funkce uvedám ve Výp. 7.10 tu, která je zavolána, pokud uživatel změní typ používané turbomolekulární vývěvy nebo ventilu. Tato `trigger` funkce nejprve zkontroluje, zda-li a jaké vakuové periferie jsou aktuálně připojeny a dále z displeje přečte a do objektu `s` uloží jejich typy. Jelikož funkce `peripheralsConnectCheck` (Výp. 7.9) z podstaty možností periferií nestanoví typ ventilu ani pumpy (tyto periferie to neumožňují) a pouze určí, zda-li jsou připojené nebo nikoliv, jejich model je vždy stanoven volbou uživatele na displeji. Funkce `stringToType` knihoven `PumpControl` a `ValveControl` slouží pro převod textového řetězce názvu pumpy či ventilu získaného čtením z displeje, na jejich model v rámci výčtu podporových periferií.

Výpis 7.10: Trigger funkce volaná při změně připojené pumpy nebo ventilu

```

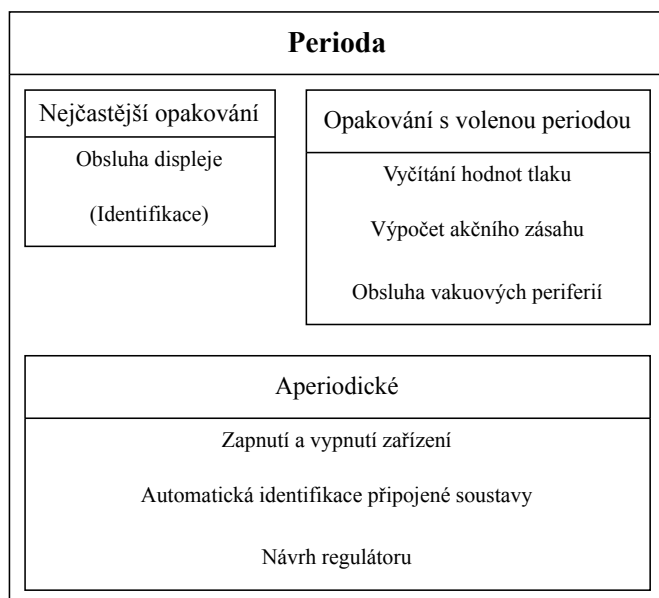
1 void trigger5() {
2     peripheralsConnectCheck();
3     s.pumpType = PumpControl::stringToType(nex.readStr("main.pump.txt"));
4     s.gauge1 = gauge::stringToType(nex.readStr("main.gauge1.txt"));
5     s.gauge2 = gauge::stringToType(nex.readStr("main.gauge2.txt"));
6     s.valveType=ValveControl::stringToType(nex.readStr("main.valve.txt"));
7     s.CH1active=s.gauge1 != gaugeNotConnected && s.gauge1 != gaugeUnused;
8     s.CH2active=s.gauge2 != gaugeNotConnected && s.gauge2 != gaugeUnused;
9 }

```

## 8 Činnosti systému

Samotná operace VSC potom probíhá v periodické smyčce, jejíž činnosti lze rozdělit do tří skupin podle rychlosti a periodicity jejich volání a ty následně do několika dalších logických bloků. Rozdělení volajících se funkcí podle jejich periodicity je vizualizováno v Obr. 8.1.

Základní činnosti zařízení jsou rozděleny podle toho, jestli k nim dochází tak často, jak použitý hardware umožňuje, nebo pouze jednou za dobu, kterou uživatel podle nároků aktuálního experimentu stanoví. Volání s tzv. *periodou CPU* (tedy co nejčastěji) se využívá pro obsluhu displeje, aby byla latence (doba mezi stiskem interaktivního prvku a patřičnou reakcí) co nejkratší a došlo tak k maximalizaci uživatelského komfortu, a také pro kontrolu přicházejících zpráv, které byly připravenými vakuovými periferiemi zaslány bez explicitního vyžádání.



Obr. 8.1: Rozdělení činností programů podle jejich periodicity

Konkrétně se sem řadí zpracování signálu z připojených měrek, čímž jsou získány aktuální hodnoty tlaku v aparatuře. Pokud je zvoleno automatické řízení tlaku, jsou stanoveny požadované hodnoty rychlosti otáčení turbomolekulární vývěvy a polohy škrtícího ventilu tak, aby bylo cílového tlaku dosaženo co nejrychleji, ale bez poškození aparatury či komponent. Následně jsou tyto povely zpracovány komunikačními obvody a poslány příslušným periferiím, které jejich příjem potvrdí zpětně zaslanou zprávou.

Mezi činnosti VSC se kromě těchto periodicky se vykonávajících řadí i další,

Do druhé kategorie potom spadají ty činnosti zařízení, které není třeba volat tak často, jak je to jen možné. Tyto činnosti zajišťují především správné řízení dynamiky vakuové aparatury, která je poměrně pomalá a aktivní zásah do ní častěji, než je to nutné, by nepřinášel pozitivní efekt. Voláním výpočetně náročných operací zajišťující pravidelnou obsluhu vakuových periferií jednou za uživatelem stanovenou dobu se tak šetří dostupný výpočetní výkon svdobách mezi voláním, čímž se zlepšuje činnost často se volajících rutin.

kteře jsou vřak vykonány pouze jednou při startu zařizení nebo aperiodicky dle požadavků uživatele. Pro zajiřtění správné funkce VSC je třeba po jeho spuřtění rozpoznat připojené periferie, pokud to jejich design dovoluje. Zařizení tak samo rozpozná připojené vakuové měřky a některé parametry turbomolekulární vývěvy. Samotný typ připojené pumpy ale nelze z komunikace s ní stanovit, a je proto třeba, aby byla pro správnou funkci zadána uživatelem ručně. Vzhledem k velikosti portfolia ventilů nabízených výrobcem, není třeba identifikaci provádět. Po stanovení modelů připojených periferií zařizení získá potřebné informace o jejich stavu a připraví je na provoz. U pumpy to může být kontrola nepřekročení limitních stavů a připravení motoru ke startu, u ventilu zase zjiřtění jeho aktuální polohy a případně přesunutí do polohy výchozí.

Činnosti, které mohou ale nemusejí být v průběhu chodu zařizení volány, jsou potom zejména identifikace vakuového systému v rámci *teorie řízení systémů* a automatický návrh na míru přizpůsobeného regulátoru, nebo ukládání dat z probíhajícího experimentu na externí médium, kterým je vložena SD karta.

Popsané činnosti, které zařizení v různých časech realizuje, nyní rozepířu detailně a také popířu, jaké zdroje popsané v Kap. 6 a Kap. 7 pro svou funkci využívají.

## 8.1 S krátkou periodou

Jak již bylo zmíněno v Kap. 5, tyto činnosti jsou volány tak často, jak je to jen možné. Tedy ihned po předchozím dokončení programové smyčky, je její část, které zahrnuje činnosti této kategorie, zavolána znovu. Jsou zde umístěny ty rutiny, jejichž časté volání je nutné nebo zásadním způsobem přínosné pro fungování VSC. Aktuálně se jedná především o zahycování zpráv zaslaných displejem, aby dořlo ke snížení prodlevy reakce zařizení na uživatelský vstup. Je dobré také uvést, že v některých okamžicích se s co nejkratří periodou vykonává i podproces identifikace zodpovědný za proměření přechodové charakteristiky pro maximalizaci dostupných vzorků. Touto problematikou se vřak více zabývám v příslušné

### 8.1.1 Obsluha displeje

Při každém opakování programové smyčky je nejprve volána metoda `NextionListen` knihovny `EasyNextion` (Sek. 7.6), která zabezpečuje zachycení a zpracování displejem posílaným zpráv. Následně je také zjiřtěna aktuálně na displeji otevřená stránka, což může způsobit volání funkcí specifických pro tuto stránku. Ty se vřak vykonávají s delří než *CPU periodou*, a jsou proto zařazeny v příslušné kapitole Sek. 8.2. Periodické kontrolování přichozích zpráv z displeje do řídící jednotky slouží také k co nejrychlejšímu volání `trigger` funkcí, jejichž příklad jsem popisoval v Výp. 7.10.

## 8.2 S dlouhou periodou

Zbylé periodicky se vykonávající činnosti jsou volány pouze, pokud od jejich posledního vykonání proběhla doba, která je uživatelem zvolena na displeji (výchozí hodnota této periody jsou tři sekundy). Vykonávají se zde také funkce specifické pro stránku grafického rozhraní displeje, na které se uživatel právě pohybuje. Například na hlavní stránce periodicky dochází k volání funkce, která stanoví hodnotu napájecího napětí. Kód této funkce, uvedený ve Výp. 8.1, využívá podobné programové struktury jako dříve vysvětlované metody a funkce v Sek. 7.6 nebo ve Výp. 7.6, a proto jej nebudu dále popisovat.

Výpis 8.1: Funkce volaná, pokud je na displeji otevřena úvodní stránka

```
1 void page0() {
2   analogReference(INTERNAL1V1);
3   supplyVoltageAn = analogRead(A2);
4   supplyVoltage = 1.1 * supplyVoltageAn / 1023 / 1790 * (45900 + 1790);
5   dataString = "Supply_voltage:_ " + String(supplyVoltage) + "_V";
6   dbSerialprintln(dataString);
7   long col = 25388;
8   if (supplyVoltage < 1) {
9     dataString = "Power_supply_not_connected!";
10    col = 49152;
11  }
12  nex.writeNum("main.supplyV.pco", col);
13  nex.writeStr("main.supplyV.txt", dataString);
14 }
```

### 8.2.1 Vyčítání hodnot tlaku

K čtení hodnot tlaku z vakuové aparatury pomocí prostředků popsanych v Sek. 7.3.2 dochází také pouze s periodou stanovenou uživatelem. Ve funkci, která toto zajišťuje, dochází také kromě volání dříve popisované knihovní metody `pressureMeas` (viz Výp. 7.7) i k ukládání aktuálních hodnot tlaku, rychlosti otáček pumpy nebo polohy ventilu na vloženou SD kartu, pokud uživatel tuto možnost na displeji povolil. V případě, že se jedná o první vykonání tohoto úkonu (v cílovém souboru pro ukládání dat na SD kartě nejsou ještě zapsány žádné hodnoty), dojde k zapsání hlavičky, která specifikuje za jakých podmínek k měření došlo – čas zahájení, typ připojených komponent, uživatelsky zadané poznámky – a jaká data byla uživatelem zvolena k ukládání.



## **8.2.2 Zpracování požadavku na cílovou hodnotu tlaku v aparatuře**

Pokud je uživatelem zvolena metoda automatické regulace tlaku, dochází v každé periodě činnosti VSC k výpočtu výstupní hodnoty regulátoru, který však před tím musí být pro danou aparaturu navržen. Tomuto procesu předchází identifikace systému vakuové aparatury a obě zmíněné činnosti popisují v příslušných Sek. 8.3.2 a Sek. 8.3.3. Výstupní hodnotou regulátoru je požadavek na rychlost otáčení turbomolekulární vývěvy. Jelikož se jedná o diskrétní způsob implementace PID regulátoru, zásadní roli na kvalitu jeho funkčnosti hraje uživatelem volená perioda systému. Čím bude zvolená perioda nižší, tím lepších výsledků regulace lze dosáhnout. Pokud chce obsluha VSC ovlivnit charakteristiku odezvy systému na změnu požadovaného hodnoty tlaku a upravit výstupy automatizovaného algoritmu návrhu regulátoru, lze na stránce nastavení v uživatelském rozhraní povolit možnost pokročilých funkcí kontroly nad systémem. Po jejím povolení je odemčena možnost ručně upravovat parametry navrženého regulátoru (zesílení a časové konstanty) nebo také měnit jeho typ (P, PI, PID).

## **8.2.3 Povelování vakuových komponent**

Jednou za uživatelem zvolenou periodu jsou rozeslány povely vakuovým periferiím s informacemi, na jakou rychlost otáček či polohu se mají nastavit. Děje se tak ať už pouhým přímým vyčtením těchto dat z displeje v případě manuální kontroly nad periferiemi, nebo výpočtem za pomoci navrženého regulátoru. V případě automatického řízení je z displeje tedy čtena pouze informace o požadované hodnotě tlaku.

## **8.3 Činnosti systému volané aperiodicky**

### **8.3.1 Zapnutí a vypnutí zařízení**

Po startu zařízení a inicializaci všech komunikačních kanálů dochází ihned ke zjištění stavu připojených vakuových komponent. Je tak možné aktualizovat prvky displeje tak, aby zobrazovaly současné údaje ještě před tím, než si je zobrazí uživatel. Upravovány jsou tak například ty objekty uživatelského rozhraní, které reprezentují aktuální rychlost otáček pumpy anebo polohu ventilu.

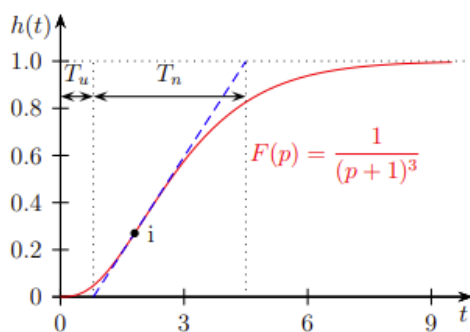
### 8.3.2 Identifikace vakuového systému

Pro regulaci tlaku ve vakuové aparatuře je třeba nejprve provést její identifikaci v rámci *teorie řízení systémů* a podle jejího výsledku automaticky navrhnout regulátor. Tuto činnost je potřeba provádět plně automaticky, tedy bez toho, aby byl kdekoliv v procesu nutný zásah obsluhy. Existuje celá řada metod, které se touto problematikou zabývají, já jsem ale potřeboval zvolit takovou, která bude prakticky realizovatelná na použitém hardwaru. [30] Identifikace systému i následný návrh regulátoru musí proběhnout přímo v zařízení a nelze tak využít specializovaných programů jako například MATLAB *System Identification Toolbox*. Na základě těchto požadavků jsem pro návrh regulátoru zvolil metodu *Ziegler-Nichols*, která pro systém navrhne regulátor na základě znalosti pouze jeho dvou parametrů – kritického zesílení  $K_u$  a periody kritických kmitů  $T_u$ . [37] Metoda návrhu regulátoru v tomto případě ovlivňuje i postup při identifikaci systému, jak bude vysvětleno dále. Ačkoliv je originální metoda formulovaná pomocí dnes se nevyužívajících parametrů, její moderní přepis shrnutý v Tab. 8.1 stanovuje postup pro návrh regulátorů typu P, PI nebo PID z kritických parametrů soustavy.

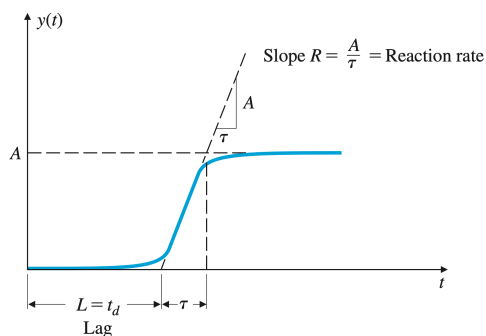
Tab. 8.1: Postup pro návrh regulátoru pomocí metody Ziegler-Nichols [10]

Typ regulátoru	$K$	$T_i$	$T_d$
P	$0,5K_u$	-	-
PI	$0,45K_u$	$\frac{T_u}{1,2} \doteq 0,83T_u$	-
PID	$0,6K_u$	$0,5T_u$	$0,125T_u$

Zmiňované parametry soustavy  $K_u$  a  $T_u$ , se kterými metoda pracuje, jsou definovány takto: kritické zesílení  $K_u$  je takové, které neregulovaný systém přivede na *mez stability* – dochází k netlumeným kmitům, které mají v čase pořád stejnou periodu.  $T_u$  je potom perioda těchto netlumených kmitů. Pro identifikaci vakuové aparatury však není vhodné do tohoto stavu systém přivádět, protože by mohlo dojít k poškození zařízení a také by celý proces trval zbytečně dlouho. Alternativní metoda pro stanovení kritických parametrů systému je schopna je stanovit z běžné přechodové charakteristiky systému, ze které jsou pomocí stanoveného zesílení systému  $k$ , doby náběhu  $T_n$  a doby průtahu  $T_p$  určeny kritické parametry. [11] Pro pochopení metody výpočtu je třeba nejdříve definovat dobu náběhu, dobu průtahu a zesílení z přechodové charakteristiky systému. Příklad pro jednoduchý systém třetího řádu je zobrazen na Obr. 8.2 nalevo.



(a) Doba náběhu a průtahu pro systém třetího řádu [8]



(b) Zesílení systému a strmost tečny k přechodové charakteristice v inflexním bodě [11]

Obr. 8.2: Parametry systému určované z přechodové charakteristiky

Pokud vedeme tečnu k přechodové charakteristice v inflexním bodu, tato tečna v určitém místě protíná časovou osu a v jiném bodě překračuje ustálenou hodnotu systému (na Obr. 8.2 vlevo překračuje hodnotu 1 normalizované přechodové charakteristiky). Čas od počátku charakteristiky po pomyslné místo průsečíku tečny a časové osy se nazývá *doba průtahu* a čas od tohoto bodu po pomyslné místo průsečíku tečny a ustálené hodnoty se zase nazývá *doba náběhu*. V Tab. 8.2 navíc figuruje parametr  $R$ , který představuje maximální strmost přechodové charakteristiky. Spočítá se prostým podílem  $R = \frac{k}{T_n}$

Porovnáním obou tabulek (Tab. 8.1 a Tab. 8.2) pro stanovení parametru PID regulátoru pomocí metody Ziegler-Nichols lze stanovit vzájemnou závislost využitých parametrů – tyto vztahy jsou uvedeny v (8.2) a (8.1).

Tab. 8.2: Postup pro návrh regulátoru pomocí metody Ziegler-Nichols při stanovení kritických parametrů z přechodové charakteristiky [11]

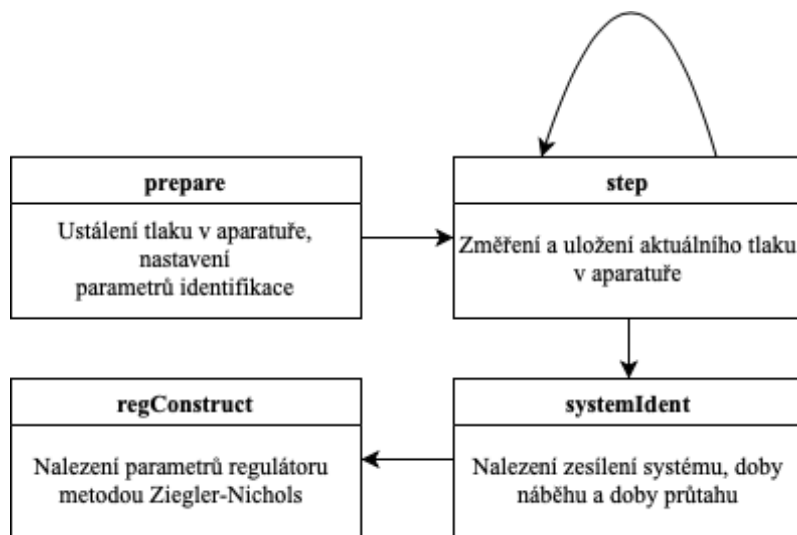
Typ regulátoru	$K$	$T_i$	$T_d$
P	$\frac{1}{RT_p}$	-	-
PI	$\frac{0,9}{RT_p}$	$\frac{T_p}{0,3}$	-
PID	$\frac{1,2}{RT_p}$	$2T_p$	$0,5T_p$

$$K_u = \frac{T_n^2}{kT_p} \quad (8.1)$$

$$T_u = 4T_p \quad (8.2)$$

Pro úspěšnou identifikaci systému jsem implementoval algoritmus, který zaručí proměření přechodové charakteristiky při skokové změně akčního zásahu, kterým jsou otáčky turbomolekulární vývěvy (viz Kap. 3). Z této charakteristiky algoritmus dále stanoví potřebné parametry – zesílení systému, dobu náběhu a dobu průtahu, z nich pomocí (8.1) a (8.2) kritické zesílení a periodu netlumených kmitů a ty předá dále pro návržení spojitého regulátoru.

Algoritmus identifikace parametrů připojeného systému jsem implementoval pomocí stavového automatu, který při svém běhu přejde celkem čtyři stavy a jeden z nich vykoná opakovaně. První stav nazvaný **prepare** slouží k inicializaci proměření soustavy – nastaví otáčky vývěvy a polohu ventilu na výchozí hodnoty a počká, až dojde k ustálení tlaku v aparatuře. Jakmile je tlak po několik period stálý, skokově se změni požadavek na otáčky turbíny a automat přechází do dalšího stavu. Ve druhém stavu **step** opakovaně dochází k zaznamenání aktuální hodnoty tlaku v aparatuře pro další výpočty. Stav je periodicky vykonáván až do té doby, kdy opět dojde k ustálení tlaku na konečné hodnotě a ten je tak po několik opakování tohoto kroku konstantní.



Obr. 8.3: Blokové schéma stavového automatu identifikace vakuové aparatury

Třetí ze čtyř stavů jsem pojmenoval jako **systemIdent** a slouží k nalezení požadovaných parametrů soustavy – zesílení, doby náběhu a doby průtahu. Posledním stavem je **regConstruct**, který z těchto parametrů pomocí metody Ziegler-Nichols

určí parametry navrhovaného regulátoru, viz Sek. 8.3.3. Přehledně jsou jednotlivé stavy automatu zobrazeny na Obr. 8.3. Softwarová implementace jednotlivých kroků je následující:

Výpis 8.2: Identifikace systému: stav `prepare`

```
1 void ident::prepare() {
2     s.regReady = false;
3     s.speedTarget = this->speedBegin;
4     s.valvePositionTarget = this->posTarget;
5
6     if (compareTol(pressurePrev, pressureActual, .5)) {
7         ++st;
8         pressureBegin = pressureActual;
9         this->index = 1;
10        sampleBegin = new sampleNode{ pressureActual, millis(), this->index
            ↪ , nullptr, nullptr };
11        sampleEnd = sampleBegin;
12        timeStart = sampleBegin->time;
13        s.speedTarget = this->speedEnd;
14    }
15    pressurePrev = pressureActual;
16 }
```

Metoda nejprve označí stav regulátoru za nepřipravený, což zabrání jeho volání z nadřazené programové smyčky, a nastaví výchozí otáčky pumpy a polohu ventilu. Dále zkontroluje, jestli je tlak od posledního volání stejný s přesností 5 %, pokud ne, uloží aktuální hodnotu tlaku a po vykonání zbytku smyčky se zavolá znovu. Pokud je ovšem tlak konstantní (došlo k jeho ustálení), zaznamená jeho výchozí hodnotu, uloží jej do oboustranně vázaného seznamu hodnot (pomocí struktury `sampleNode`), skokově změní požadavek na otáčky turbomolekulární vývěvy a inkrementuje proměnnou `st`, která ukládá aktuálně se vykonávající stav automatu.

Výpis 8.3: Identifikace systému: stav `step`

```
1 void ident::step() {
2     log();
3     if (index>5) {
4         if (constPressure(index-3, index, .5)) {
5             pressureEnd = pressureActual;
6             ++st;
7         }
8     }
9 }
```

Metoda `step` provede uložení aktuálních hodnot tlaku do seznamu struktur zavoláním `log`, tak jako tomu bylo při začátku měření ve stavu `prepare` Výp. 8.2. K přechodu do dalšího stavu dochází ve chvíli, pokud je tlak konstantní po dobu pěti posledních period, což zaručuje metoda `constPressureRecent` s argumentem vykonaných period, po které mají být tlaky s tolerancí 5 % konstantní. Uloží se konečná hodnota tlaku a automat přechází do dalšího stavu.

#### Výpis 8.4: Identifikace systému: stav `systemIdent`

```

1 void ident::systemIdent() {
2     if(sampleBegin == nullptr || sampleEnd == nullptr) return;
3     uint16_t ixInfl = inflexion(), i = 0;
4     unsigned long timeBegin, timeEnd = 0, timeSecant;
5     sampleNode* node = sampleBegin, *nodeInfl;
6
7     // time begin
8     timeBegin = timeStart;
9     sampleBegin->time = timeStart;
10
11    // time inflex 1
12    while(true) {
13        if (i++ == ixInfl) {
14            nodeInfl = node;
15            break;
16        }
17        node = node->next;
18    }
19
20    // time secant
21    double pressureDiff = (nodeInfl->next->pressure - nodeInfl->prev->
    ↪ pressure) / 2;
22    double timeDiff = double(nodeInfl->next->time - nodeInfl->prev->time)
    ↪ / 2;
23    timeSecant = double(nodeInfl->time) - (nodeInfl->pressure -
    ↪ sampleBegin->pressure) / pressureDiff * timeDiff;
24
25    // time End
26    timeEnd = double(nodeInfl->time) + (sampleEnd->pressure - nodeInfl->
    ↪ pressure) / pressureDiff * timeDiff;
27
28    riseTime = double(timeSecant)/1000;
29    peakTime = double(timeEnd)/1000 - double(timeSecant)/1000;

```

```

30     double Xpercent = (speedEnd - speedBegin) / (s.maxSpeed - s.minSpeed);
31     gain = Xpercent * (pressureEnd - pressureBegin);
32
33     Ku = double(peakTime) * double(peakTime) / (gain * double(riseTime));
34     Tu = 4 * double(riseTime);
35
36     ++st;
37     s.speedTarget = this->speedBegin;
38 }

```

Metoda `systemIdent` po ukončení měření stanoví dobu náběhu, průtahu a zesílení systému a z nich kritické parametry podle (8.2) a (8.1). Pro správnou funkci algoritmu je nejprve potřeba nalézt inflexní bod diskrétní přechodové charakteristiky. Hodnotu charakteristiky v inflexním bodě stanoví funkce `inflex()`, která tuto hodnotu nalezne pomocí vyhledání změny znaménka druhé diference (jedná se o diskrétní funkci). Pro hodnotu inflexního bodu je dále nalezen příslušný uzel `sampleNode` a z něj je pomocí hodnoty tlaku v okolních uzlech stanovena pomyslná směrnice tečny k tomuto bodu. Metoda nakonec určí, v jakém čase je hodnota této tečny rovna počátečnímu tlaku a v jakém ustálené hodnotě přechodové charakteristiky (tedy koncovou hodnotu tlaku). Z nich je již snadné stanovit dobu náběhu a dobu průtahu a ze znalosti velikosti akčního zásahu, který odpovídá změně požadavku na otáčky pumpy ze stavu `prepare` Výp. 8.2, a rozdílu počátečního a koncového tlaku také zesílení. Podle vzorců (8.1) a (8.2) jsou konečně stanoveny příslušné hodnoty kritického zesílení a periody netlumených kmitů. Po stanovení všech parametrů automat přechází do dalšího stavu, ve kterém dochází k návrhu regulátoru.

### 8.3.3 Návrh regulátoru

Regulaci tlaku v aparatuře jsem implementoval tak, aby akční zásah, který je výstupem regulátoru, odpovídal požadavku na rychlost otáček turbomolekulární vývěvy. Vakuový ventil, jako druhý akční prvek

Stanovení parametrů uživatelem zvoleného regulátoru probíhá v posledním stavu implementovaného stavového automatu, jehož předcházející činnost popisují v sekci Sek. 8.3.2. Posledním stavem je `regConstruct`, který pouze z hodnot kritického zesílení a periody netlumených kmitů dle vztahů z Tab. 8.1 určí hodnoty  $K$ ,  $T_I$  a  $T_D$  regulátoru. Po dokončení výpočtu metoda změní stav regulátoru na „připraven“ a automat se vrátí do svého původního stavu a je tak připraven na případnou další identifikaci. Dochází také k uvolnění paměti, kterou dosud zabíral vázaný seznam s naměřenými hodnotami tlaku.

### Výpis 8.5: Identifikace systému: stav regConstruct

```

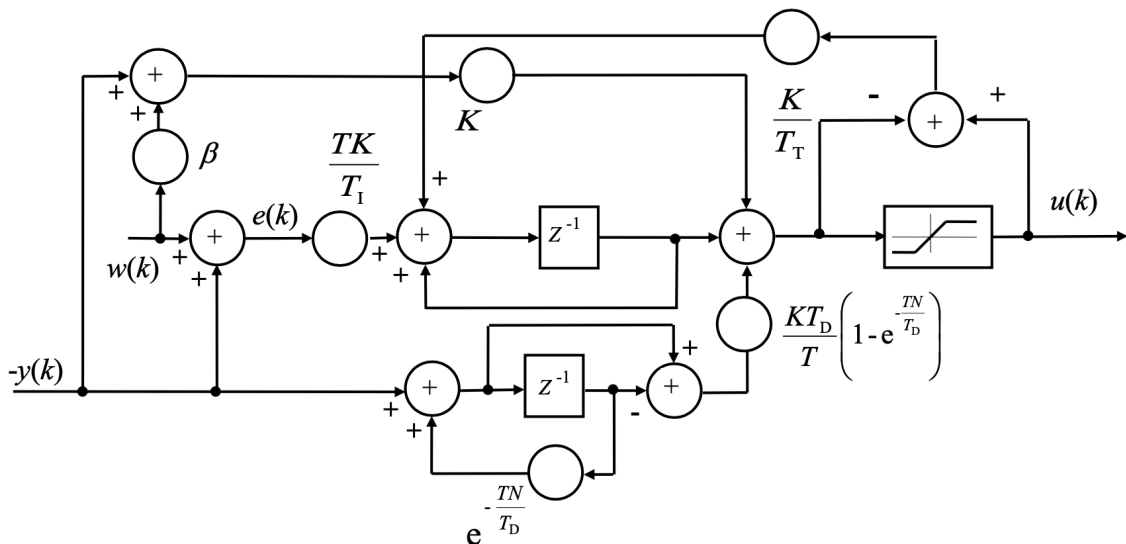
1 void ident::regConstruct() {
2     if (target == gauge1 || target == both) {
3         delete Reg;
4         if (setRegType == PSD) {
5             Reg = new reg{0.6 * Ku, 0.5 * Tu, 0.125 * Tu, N, s.period,
6                 ↪ setRegType, s};
7         }
8         else if (setRegType == PS)
9             Reg = new reg{0.45 * Ku, Tu / 1.2, s.period, s};
10        else
11            Reg = new reg{0.5 * Ku, s.period, s};
12    }
13    else {
14        delete Reg2;
15        if (setRegType == PSD)
16            Reg2 = new reg{0.6 * Ku, 0.5 * Tu, 0.125 * Tu, N, s.period,
17                ↪ setRegType, s};
18        else if (setRegType == PS)
19            Reg2 = new reg{0.45 * Ku, Tu / 1.2, s.period, s};
20        else
21            Reg2 = new reg{0.5 * Ku, s.period, s};
22    }
23    s.regReady = true;
24    s.period = periodBefore;
25    s.identOn = false;
26    st = 0;
27
28    auto node = sampleBegin;
29    auto tmp = node;
30    while(node->next != nullptr) {
31        node = node->next;
32        delete tmp;
33        tmp = node;
34        ++u;
35    }
36    delete node;
37 }

```

Jak jsem ukázal v předchozí sekci Sek. 8.3.2, z naměřené přechodové charakteristiky lze určit kritické parametry aktuálně připojené vakuové aparatury a z nich následně jednoduchým postupem navrhnout spojitý regulátor. VSC však, tak jako



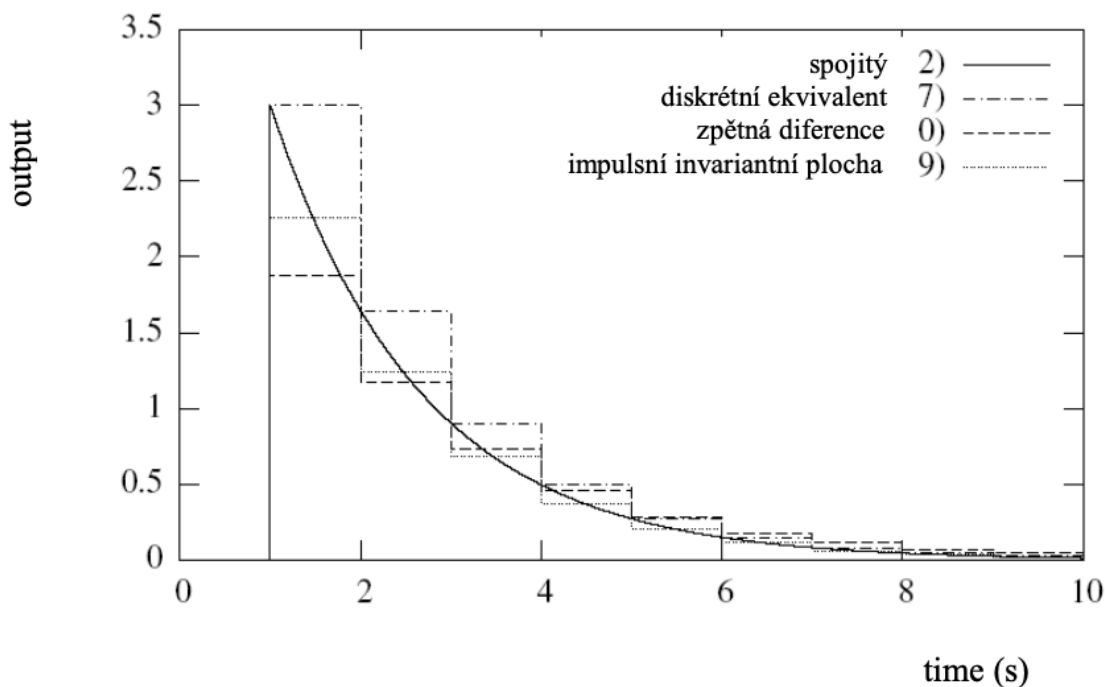
každý moderní systém řízení, pracuje v diskrétním čase a tedy nikoliv spojitě. Posledním krokem pro úspěšný návrh funkčního regulátoru je tedy nalezení diskrétního ekvivalentu ke spojitě verzi získané pomocí metody Ziegler-Nichols. Aby bylo umožněno plynulé přepínání mezi automatickým řízením, kdy je tlak regulován regulátorem, a manuálním ovládáním, kdy rychlost vývěvy a polohu ventilu přímo stanovuje uživatel, zvolil jsem implementaci regulátoru s *beznárazovým přepínáním*. Zvolená implementace diskrétního řízení uvedena na Obr. 8.4 a kromě beznárazového přepínání obsahuje také množství dalších funkčních bloků. Omezení překmitu regulované veličiny zajišťuje úprava klasického diskrétního PSD regulátoru zvaná  $\beta$ -PSD [30]. Přesaturování integrátoru (sumátoru) zase zabraňuje blok *anti wind-up*, který v případě saturace výstupní hodnoty regulátoru (dosažení maximálních nebo naopak minimálních otáček vývěvy) „odebírá“ z naintegrované hodnoty. Derivační (diferenční) složka je filtrována metodou invariantní plochy, která oproti jiným metodám aproximace spojitě derivace poměrně věrně kopíruje její průběh, viz Obr. 8.5.



Obr. 8.4: Blokové schéma implementovaného diskrétního regulátoru [30]

Konstanty vyskytující se v blokovém schématu regulátoru na Obr. 8.4 reprezentují následující:

- $y(k)$  – výstupní hodnota soustavy
- $u(k)$  – vstup do soustavy, výstup regulátoru
- $w(k)$  – požadovaná hodnota
- $e(k)$  – regulační odchylka
- $T$  – vzorkovací perioda
- $K$  – zesílení regulátoru
- $T_I$  – časová konstanta integrátoru
- $T_D$  – časová konstanta derivační složky



Obr. 8.5: Srovnání metod filtrací derivační složky [30]

$N$  – realizační konstanta

$T_T$  – *anti wind-up* konstanta

$\beta$  – parametr regulátoru omezující překmit regulované veličiny

Požadovaná a výstupní hodnota jsou údaje o tlaku, kterého chceme dosáhnout a který je aktuálně v aparatuře, výstup regulátoru též nazývaný jako akční zásah jsou otáčky turbomolekulární vývěvy. Vzorkovací perioda je doba opakování smyčky pomaleji se vykonávajících činností a je, stejně jako požadovaný tlak, dána vstupem uživatele, viz Kap. 8. Zesílení regulátoru, časová konstanta integrátoru a časová konstanta derivační složky jsou parametry získané pomocí metody Ziegler-Nichols, kterou jsem popisoval v Sek. 8.3.3. Realizační konstanta byla stanovena experimentálně a jedná se o parametr vyskytující se ve výpočtu diskrétního ekvivalentu derivace (diference) – běžnou derivaci v aktuálním bodě není možné stanovit, protože ještě není znám další vývoj derivované funkce. *Anti wind-up konstanta* ovlivňuje rychlost, se kterou je měněna naintegrovaná hodnota v případě, kdy dochází k saturaci výstupu. Stanovena byla opět experimentálně, stejně jako poslední parametr  $\beta$ , který je specifikem této verze PSD regulátoru, a slouží k omezování vlivu požadované hodnoty na proporční P složky regulátoru. Hodnoty všech parametrů jsou uvedeny v Tab. 8.3.

Regulátor jsem implementoval opět jako knihovnu, jejíž objekt představuje navržený regulátor. Vnitřní proměnné objektu ukládají informace jak o typu regulátoru, tak o hodnotách jeho parametrů a pro účely výpočtu akčního zásahu i hodnoty

některých jeho minulých stavů. Kód regulátoru na Výp. 8.6 (viz Výp. 8.6) je programovým přepisem blokového schématu z Obr. 8.4 a funguje následovně: nejprve je stanovena aktuální hodnota tlaku  $y$  a požadovaná hodnota tlaku  $w$ , která je převedena na stejnou jednotku (Pa). Dále jsou spočteny hodnoty jednotlivých složek regulátoru PSD, případně jiného typu. Součet dílčích složek je saturován podle minimálních a maximálních otáček turbomolekulární vývěvy a do proměnných  $h_d$  a  $h_s$  jsou uloženy hodnoty pro výpočet D a S složky při dalším volání metody.

Tab. 8.3: Souhrnný přehled parametrů použitého  $\beta$ -PSD regulátoru

Parametr	Hodnota
$T$	dána uživatelem (výchozí hodnota 3 s)
$K$	pomocí metody Ziegler-Nichols
$T_I$	pomocí metody Ziegler-Nichols
$T_D$	pomocí metody Ziegler-Nichols
$N$	100
$T_T$	50000
$\beta$	0,5

### Výpis 8.6: Regulátor: periodický výpočet akčního zásahu

```
1 double reg::tick() {
2     double w = s.targetPressureVal, y = s.pressure1;
3     if (s.targetPressureUnit == mPa) w /= 1000;
4     if (s.targetPressureUnit == uPa) w /= 1000;
5     double P = K * (beta * w - y);
6     double D = (type == PSD) ? invariant * (expV * h_d - h_d - y) : 0;
7     double S = (type == P) ? 0 : h_s;
8     res = S + P + D;
9     if (res > satMax) res = satMax;
10    if (res < satMin) res = satMin;
11    h_d = h_d * expV - y;
12    h_s = S + T * K / Ti * (w-y) + K / Tt * (- res + (S + P + D));
13    return res;
14 }
```

## 9 Validace automatické identifikace a regulace

Při ověřování správnosti implementace algoritmu identifikace soustavy a návrhu regulátoru jsem narazil na výkonové limitace aktuálního řešení. Programová smyčka obsahující pouze nutné funkce pro správný průběh identifikace se vykonává s periodou rovnou přibližně jedné sekundě. Při umístění aktuálního pracovního bodu

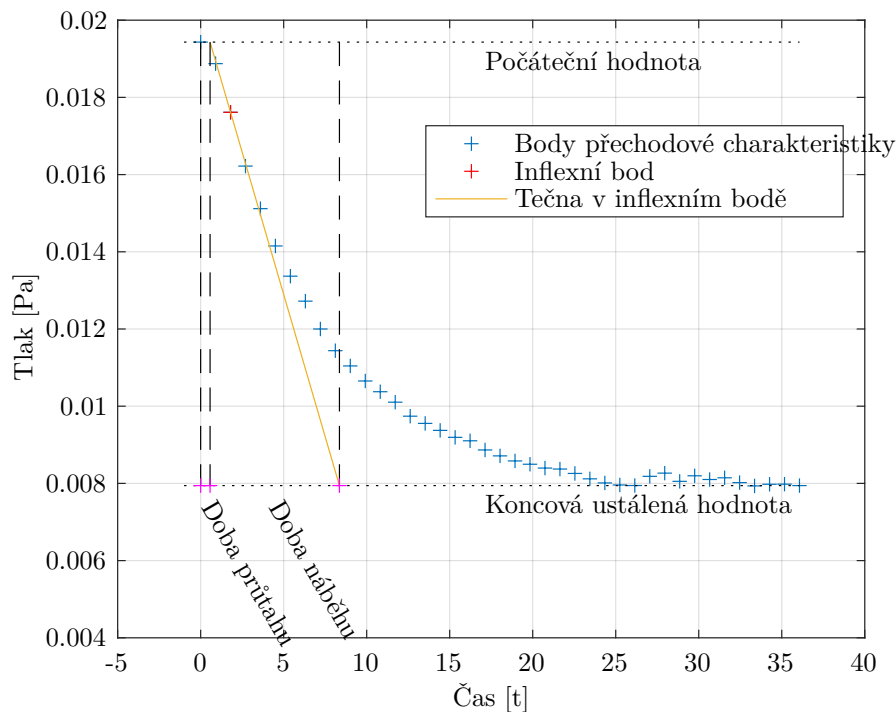
Tab. 9.1: Naměřená identifikační data pro počáteční tlak  $p_0 = 19,432$  mPa a akční zásah  $750 \rightarrow 1500$  Hz

t [s]	p [mPa]	t [s]	p [mPa]
0.000	19.432	18.935	8.582
0.900	18.874	19.837	8.496
1.800	17.614	20.739	8.397
2.700	16.220	21.640	8.372
3.601	15.117	22.541	8.258
4.502	14.151	23.443	8.118
5.404	13.368	24.345	8.012
6.307	12.721	25.247	7.959
7.210	12.000	26.149	7.945
8.112	11.438	27.051	8.183
9.013	11.043	27.952	8.266
9.915	10.653	28.853	8.053
10.817	10.375	29.755	8.197
11.719	10.104	30.657	8.101
12.621	9.741	31.559	8.145
13.523	9.554	32.461	8.020
14.426	9.373	33.363	7.935
15.327	9.193	34.266	7.977
16.228	9.104	35.167	7.980
17.130	8.866	36.068	7.943
18.032	8.713		

vývěvy do vyšších tlaků pomocí většího otevření vakuového ventilu je přechodový děj i při maximální možné skokové změně akčního zásahu (skok z minimálních otáček 750 Hz na maximální 1500 Hz) příliš rychlý na to, aby bylo možné algoritmem získat věrohodná data. Při zpomalení přechodového děje větším přivřením ventilu je perioda zachycení dat dostatečná pro návrh funkčního regulátoru. Data naměřená za počátečních podmínek  $p_0 = 19,432$  mPa a  $u = 750$  Hz jsou známenána v Tab. 9.1 a také společně s ilustrací algoritmu výpočtu kritických parametrů zobrazena v Obr. 9.1. Výsledné parametry stanovené identifikační metodou `systemIdent()` popisanou v Sek. 8.3.2 jsou následující: doba průtahu  $T_p = 0,567$  s, doba náběhu  $T_n = 8,359$  s, zesílení  $k = -0,0115$  a poté kritické parametry určené metodou podle (8.1) a (8.2) – kritické zesílení  $K_u = -10715,88$  a perioda netlučených kmitů  $T_u = 2,268$  s. Proměnné parametry z těchto hodnot určeného  $\beta$ -PSD regulátoru tak podle Tab. 8.1 jsou: zesílení  $K = -6429,5$ , integrační časová konstanta  $T_i = 1,134$  s a derivační ča-

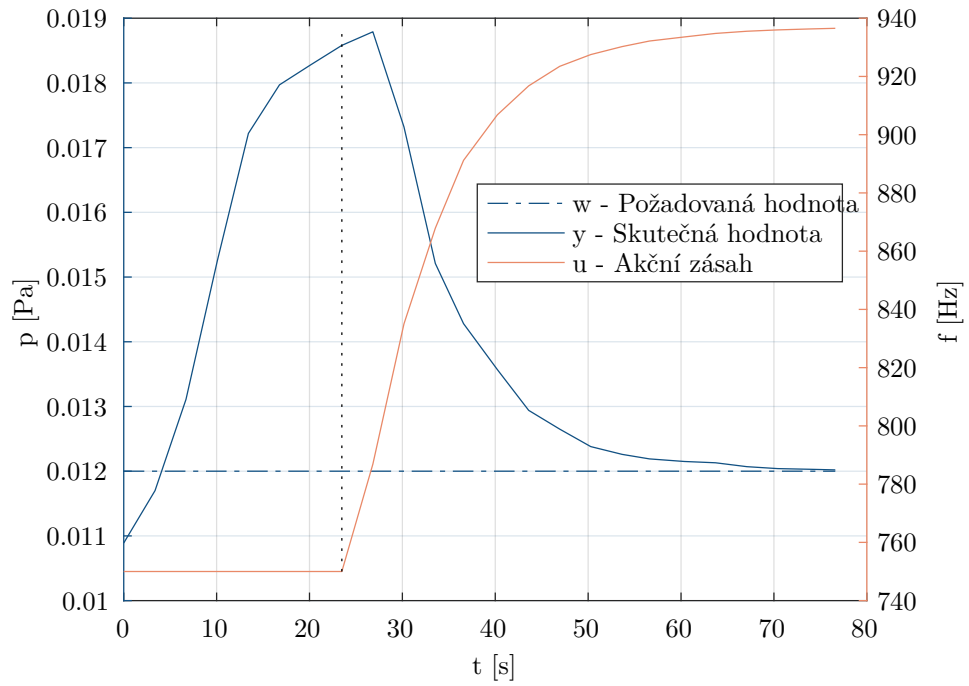
sová konstanta  $T_d = 0,284$  s. Zbylé parametry regulátoru nejsou předmětem návrhu pomocí metody Ziegler-Nichols a zůstávají neměnné s hodnotami dle Tab. 8.3.

Pro regulátor s takto nastavenými parametry byl při minimálních otáčkách turbíny zadán požadavek na dosažení tlaku  $p = 12 \text{ mPa}$ . Průběh této regulace, velikost akčního zásahu i jednotlivých složek regulátoru je zobrazen v Tab. 9.2. Vývoj akčního zásahu i skutečné hodnoty tlaku v aparatuře v čase je dobře viditelný na Obr. 9.2. Prvních přibližně 25 sekund tohoto grafu zůstávají otáčky turbíny beze změny a posléze začínají narůstat. O chvíli později (v další periodě programu) již hodnota tlaku v aparatuře začíná na tuto změnu reagovat a postupně dochází k eliminaci regulační odchylky. Ta je po ca 70 sekundách od začátku regulace prakticky rovna nule. K zásadnímu zrychlení regulačního děje by mohlo dojít, pokud by regulátor začal měnit otáčky turbomolekulární vývěvy ihned po začátku regulace a nikoliv až se zpožděním.

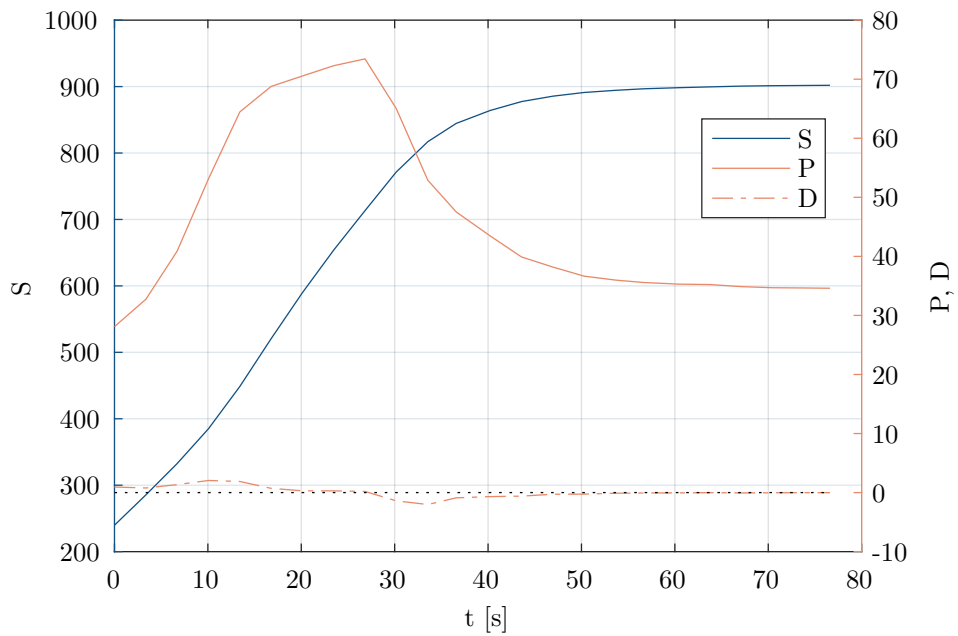


Obr. 9.1: Identifikace reálného systému

Důvod této prodlevy je dobře patrný na Obr. 9.3, kde je proti levé vertikální ose zobrazena hodnota S složky regulátoru v čase a proti pravé vertikální ose zase P a D složka. Je zřejmé, že sumační složka akčního zásahu tvoří jeho převážnou většinu. Součet všech tří složek je ovšem na začátku regulačního děje výrazně menší, než je minimální možný akční zásah, tedy  $u_{min} = 750 \text{ Hz}$  daný minimálními otáčkami vývěvy. Požadavek pumpě na rychlost jejího otáčení tedy zůstává na minimální rychlosti (vlivem saturace na výstupu regulátoru) až do momentu, kdy součet všech tří složek přesáhne oní minimální hodnotu 750 Hz v čase přibližně 25 sekund.



Obr. 9.2: Akční zásah regulátoru



Obr. 9.3: Vývoj složek S, P a D regulátoru

Tab. 9.2: Vstupy, stavy a výstupy regulátoru při  $w = 12 \text{ mPa}$ ,  $T = 3 \text{ s}$

<b>t[s]</b>	<b><math>u</math> [Hz]</b>	<b>S</b>	<b>P</b>	<b>D</b>	<b><math>w</math> [mPa]</b>	<b><math>y</math> [mPa]</b>
0,000	750,00	239,77	28,10	0,94	12,00	10,89
3,371	750,00	285,55	32,75	0,78	12,00	11,70
6,704	750,00	332,50	40,87	1,35	12,00	13,11
10,075	750,00	385,17	53,16	2,05	12,00	15,25
13,423	750,00	448,64	64,46	1,88	12,00	17,22
16,767	750,00	520,42	68,77	0,72	12,00	17,97
20,135	750,00	590,01	70,55	0,30	12,00	18,28
23,483	750,00	654,10	72,29	0,29	12,00	18,58
26,823	786,87	713,24	73,44	0,19	12,00	18,79
30,170	835,03	771,41	65,03	-1,40	12,00	17,32
33,546	867,90	817,03	52,90	-2,02	12,00	15,21
36,585	891,18	844,54	47,53	-0,89	12,00	14,28
40,225	906,82	864,04	43,46	-0,68	12,00	13,57
43,597	916,76	877,47	39,89	-0,60	12,00	12,94
46,937	923,47	885,56	38,19	-0,28	12,00	12,65
50,280	927,51	891,12	36,65	-0,26	12,00	12,38
53,658	930,25	894,38	35,98	-0,11	12,00	12,26
56,596	932,16	896,64	35,58	-0,07	12,00	12,19
60,356	933,56	898,31	35,30	-0,05	12,00	12,15
63,713	934,77	899,56	35,23	-0,01	12,00	12,13
67,085	935,52	900,70	34,88	-0,06	12,00	12,07
70,409	935,98	901,32	34,69	-0,03	12,00	12,04
73,740	936,30	901,66	34,65	-0,01	12,00	12,03
76,607	936,53	901,94	34,60	-0,01	12,00	12,02



# Závěr

V bakalářské práci jsem podle bodů zadání navrhnul a realizoval systém pro měření a řízení tlaku ve vakuové aparatuře. V této činnosti jsem vycházel z teoretických poznatků a znalostí nabitých při vypracovávání rešerši vakuvých komponent a analogově-digitálních převodníků. V první zmiňované rešerši jsem se zaměřil na typy používaných vakuových měrek v Sek. 1.1, dále jsem rozebral možné přístupy k realizaci odčerpávání plynu z aparatury pro dosažení požadované malých tlaků v Sek. 1.2 a nakonec shrnul možnosti propojení více takovýchto vakuových systémů pomocí ventilů v Sek. 1.3. Ve druhé rešerši jsem provedl průzkum používaných převodníků pro převod spojitě napěťové hodnoty na hodnotu číslicovou v Kap. 2, čehož jsem následně využil při zpracování měřených hodnot tlaku.

Dále jsem v Kap. 3 provedl návrh systému řízení vakuové aparatury a stanovil jsem, jaké součástky budu mít při řešení k dispozici a jaké mohu libovolně volit a dále jsem určil některé požadavky na celý řídicí systém. Tyto jsou graficky shrnuty v Obr. 3.1, který shrnuje všechny komponenty systému, jejich funkci a vzájemné propojení.

V praktické části práce jsem navržený systém realizoval jako kompaktní počítač, který po připojení k vakuové aparatuře a jejím periferiím realizuje požadovanou funkci měření a řízení tlaku. Jednotlivé hardwarové i softwarové komponenty, které jsem pro splnění tohoto zadání vytvořil, jsou uvedeny v příslušných Kap. 6 a Kap. 7. Komunikaci mezi obsluhou zařízení a zařízením samotným jsem realizoval pomocí velkého dotykového displeje zabudovaného do těla zařízení, pro který jsem také navrhnul vhodné uživatelské rozhraní (viz Sek. 5.1).

Vyhotovené řízení tlaku v aparatuře buď probíhá v manuálním režimu, kdy uživatel přímo poveluje vakuové komponenty realizující akční zásah, nebo v režimu automatickém, kdy po provedení identifikace soustavy (viz Sek. 8.3.2) dochází k regulování na požadovanou hodnotu tlaku pomocí turbomolekulární vývěvy. Tento proces je popsán v Sek. 8.3.3. Funkčnost těchto algoritmů jsem ověřil a zhodnotil v Kap. 9.

Byť bylo zadání realizováno, na vývoji zařízení plánuji dále pokračovat a v budoucnu se tak zaměřit na některé nedostatky aktuálního řešení. Zejména se jedná o migraci projektu na výkonnější centrální jednotku, která umožní lepší identifikace soustavy i plynulejší všeobecný chod zařízení. Další vylepšení plánuji pro systém regulace tak, aby bylo požadované hodnoty dosaženo rychleji a aby byl do procesu zahrnut i vakuový ventil.

# Literatura

- [1] Cold Cathode Vacuum Gauge PKR251 Product Datasheet. <https://static.pfeiffer-vacuum.com/productPdfs/PTR26000.en.pdf>, [online], [vid. 10. 03. 2023].
- [2] EVR 116 Motorized Valve Product Datasheet. <https://static.pfeiffer-vacuum.com/productPdfs/PFI39931A.en.pdf>, [online], [vid. 10. 03. 2023].
- [3] HiPace 30 with TC 110 Product Datasheet. <https://static.pfeiffer-vacuum.com/productPdfs/PMP05280.en.pdf>, [online], [vid. 10. 03. 2023].
- [4] HiPace 80 with TC 110 Product Datasheet. <https://static.pfeiffer-vacuum.com/productPdfs/PMP03940.en.pdf>, [online], [vid. 10. 03. 2023].
- [5] Pirani Vacuum Gauge TPR 270 Product Datasheet. <https://static.pfeiffer-vacuum.com/productPdfs/PTR26770.en.pdf>, [online], [vid. 10. 03. 2023].
- [6] ARDUINO: Arduino.cc. [online], [vid. 26. 12. 2022]. Dostupné z: <https://arduino.cc/>
- [7] ARDUINO; José, B.; Taddy, C.: Arduino Memory Guide. 12 2022, [online], [vid. 26. 12. 2022]. Dostupné z: <https://docs.arduino.cc/learn/programming/memory-guide>
- [8] BLAHA, P.; VAVŘÍN, P.: Řízení a regulace 1. Skriptum VUT.
- [9] Davide, B.: *Analog Electronics for Measuring Systems*, kapitola 4. John Wiley & Sons, Ltd, 2017, ISBN 9781119405177. Dostupné z: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119405177.ch4>
- [10] Ellis, G.: Chapter 6 - Four Types of Controllers. In *Control System Design Guide (Fourth Edition)*, editace G. Ellis, Boston: Butterworth-Heinemann, fourth edition vydání, 2012, ISBN 978-0-12-385920-4, s. 97–119.
- [11] FRANKLIN, G. F.; POWELL, J.; EMAMI-NAEINI, A.: *Feedback Control of Dynamic Systems*, kapitola 4.3.6. Ziegler–Nichols Tuning of the PID Controller. Essex, VB: Pearson Education Limited, 7 vydání, 2015, ISBN 978-1-29-206890-9, s. 227–229.

- [12] JOUSTEN, K. (editor): *Handbook of Vacuum Technology*, kapitola 13. Total Pressure Vacuum Gauges. Wiley-VCH Verlag GmbH, 2016, ISBN 978-3-527-68825-8.
- [13] JOUSTEN, K. (editor): *Handbook of Vacuum Technology*, kapitola 7. Positive Displacement Pumps. Wiley-VCH Verlag GmbH, 2016, ISBN 978-3-527-68825-8.
- [14] JOUSTEN, K. (editor): *Handbook of Vacuum Technology*, kapitola 9. Jet and Diffusion Pumps. Wiley-VCH Verlag GmbH, 2016, ISBN 978-3-527-68825-8.
- [15] JOUSTEN, K. (editor): *Handbook of Vacuum Technology*, kapitola 10. Molecular and Turbomolecular Pumps. Wiley-VCH Verlag GmbH, 2016, ISBN 978-3-527-68825-8.
- [16] JOUSTEN, K. (editor): *Handbook of Vacuum Technology*, kapitola 12. Cryotechnology and Cryopumps. Wiley-VCH Verlag GmbH, 2016, ISBN 978-3-527-68825-8.
- [17] JOUSTEN, K. (editor): *Handbook of Vacuum Technology*, kapitola 17.4.3 Valves. Wiley-VCH Verlag GmbH, 2016, ISBN 978-3-527-68825-8.
- [18] JOUSTEN, K. (editor): *Handbook of Vacuum Technology*, kapitola 11. Sorption Pumps. Wiley-VCH Verlag GmbH, 2016, ISBN 978-3-527-68825-8.
- [19] KONDAVEETI, H. K.; KUMARAVELU, N. K.; VANAMBATHINA, S. D.; aj.: A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. *Computer Science Review*, ročník 40, 2021, ISSN 1574-0137. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S1574013721000046>
- [20] KYCON: *KPJX Series Product Datasheet*. 2013.
- [21] MEANWELL: *220W AC-DC High Reliability Medical Adaptor*. 2016, gSM220A Series Product Datasheet.
- [22] Microchip: *MCP3422/3/4 Product Datasheet*. [online], [vid. 26. 12. 2022]. Dostupné z: <https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/22088c.pdf>
- [23] MOLEX: [molex.com](https://molex.com). [online], [vid. 14.5.2023]. Dostupné z: <https://cz.farnell.com/molex/45135-0601/cable-assy-6pos-rcpt-rcpt-150mm/dp/2723787?st=molex%20mini%20fit>

- [24] NEXTION: nextion.tech. [online], [vid. 26. 12. 2022]. Dostupné z: <https://nextion.tech/>
- [25] NXP: *PCF8523 Real-Time Clock (RTC) and calendar*. 2015, [online], [vid. 14.5.2023].
- [26] Pfeiffer Vacuum: *Cold Cathode Vacuum Gauge PKR251 Product Datasheet*. [online], [vid. 26. 12. 2022]. Dostupné z: <https://static.pfeiffer-vacuum.com/productPdfs/PTR26000.en.pdf>
- [27] Pfeiffer Vacuum: *HiPace 10 with TC 110 Product Datasheet*. [online], [vid. 26. 12. 2022]. Dostupné z: <https://static.pfeiffer-vacuum.com/productPdfs/PMP03960.en.pdf>
- [28] Pfeiffer Vacuum: *Pirani Vacuum Gauge TPR280 Product Datasheet*. [online], [vid. 26. 12. 2022]. Dostupné z: <https://static.pfeiffer-vacuum.com/productPdfs/PTR26950.en.pdf>
- [29] Pfeiffer Vacuum: *PKR 36X & IKR 36X Pirani/Cold Cathode Operating Instructions*. [online], [vid. 17. 5. 2023]. Dostupné z: [https://www.pfeiffer-vacuum.com/filepool/file/measurement/bg5164ben\\_c\\_ikr-36x.pdf](https://www.pfeiffer-vacuum.com/filepool/file/measurement/bg5164ben_c_ikr-36x.pdf)
- [30] PIVOŇKA, P.: Číslicová řídicí technika. Skriptum VUT, říjen 2012.
- [31] Renesas: *ISL8485, ISL8490, ISL8491 Product Datasheet*. 2021, 5V, Low Power, High Speed, RS-485/RS-422 Transceivers, FN6046.
- [32] Technology, P.: powerstream.com. [online], [vid. 17.5.2023]. Dostupné z: [https://www.powerstream.com/Wire\\_Size.htm](https://www.powerstream.com/Wire_Size.htm)
- [33] Texas Instruments: *ADS131M02 Product Datasheet*. [online], [vid. 26. 12. 2022]. Dostupné z: <https://www.ti.com/lit/ds/symlink/ads131m02.pdf?HQS=dis-mous-null-mouser-mode-dsf-pf-null-ww&ts=1672821690709>
- [34] Texas Instruments: *MAX232x Dual EIA-232 Drivers/Receivers Product Datasheet*. 2014, sLLS047M.
- [35] Texas Instruments: *LM741 Operational Amplifier Product Datasheet*. 2015, sNOSC25D.
- [36] Texas Instruments: *OPAx189 Precision, Lowest-Noise, 36-V, Zero-Drift, 14-MHz, MUX-Friendly, Rail-to-Rail Output Operational Amplifiers Product Datasheet*. 2021, sBOS830I.

- [37] ZIEGLER, J. G.; NICHOLS, N. G.: Optimum Settings for Automatic Controllers. *American Society for Mechanical Engineers*, ročník 64, 1942: s. 759–768.