

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV MIKROELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF MICROELECTRONICS

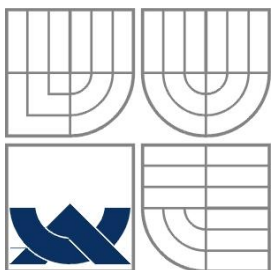
DÁLKOVĚ ŘÍZENÉ PRACOVÍŠTĚ ELEKTROCHEMICKÝCH
SENZORŮ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

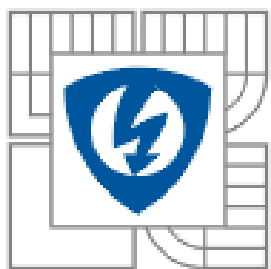
Bc. JAKUB CIESLAR

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV MIKROELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF MICROELECTRONICS

DÁLKOVĚ ŘÍZENÉ PRACOVISTĚ ELEKTROCHEMICKÝCH SENZORŮ

THE REMOTE CONTROL WORKPLACE OF ELECTROCHEMICAL SENSORS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAKUB CIESLAR

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MARTIN ADÁMEK, Ph.D.

BRNO 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav mikroelektroniky

Diplomová práce

magisterský navazující studijní obor
Mikroelektronika

Student: Bc. Jakub Cieslar
Ročník: 2

ID: 73073
Akademický rok: 2010/2011

NÁZEV TÉMATU:

Dálkově řízené pracoviště elektrochemických senzorů

POKyny PRO VYPRACOVÁNÍ:

Seznamte se základními měřicími metodami v oblasti elektrochemických senzorů. Seznamte se s jazyky pro tvorbu zabudovaného programového vybavení (embedded software), aplikací pro operační systém Windows a aplikací pro tvorbu internetového rozhraní (např. C, PHP, HTML). Navrhněte a realizujte pracoviště elektrochemických senzorů, které bude řízeno v rámci sítě WAN, případně LAN.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

Termín zadání: 7.2.2011

Termín odevzdání: 26.5.2011

Vedoucí práce: Ing. Martin Adámek, Ph.D.

prof. Ing. Vladislav Musil, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

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

Abstrakt:

Předkádaná diplomová práce se zabývá řešením automatizovaného pracoviště pro měření s elektrochemickými senzory. Hlavním cílem práce je návrh a realizace zařízení pro voltametrická měření, které je primárně určeno pro měření s tlustovrstvými elektrochemickými senzory, které komunikuje s počítačem a které je možné ovládat v rámci sítí WAN, případně LAN. Práce popisuje postupy při návrhu, realizaci a optimalizaci zařízení. Dále popisuje vytvořené programové vybavení zařízení, ovladače pro obslužný počítač a realizovaný internetový uživatelský systém.

Klíčová slova:

Internet, vzdálená správa, elektrochemické senzory, ATmega128, WAN, LAN, komunikace.

Abstract:

This master's thesis deals with the design of automated workplace for measurement with electrochemical sensors. Design and realization of a device, which communicates with computer and which can be controlled via WAN or LAN nets, are the main aims of this work. This thesis describes the process of designing, realization and optimalization of designed preparation adjustment. It also describes created software of a device, driver for service PC and programme of a web client.

Keywords:

Internet, remotely control, electrochemical sensors, ATMega128, WAN, LAN, communication.

Bibliografická citace díla:

CIESLAR, J. *Dálkově řízené pracoviště elektrochemických senzorů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 70 s. Vedoucí diplomové práce Ing. Martin Adámek, Ph.D.

Prohlášení autora o původnosti díla:

Prohlašuji, že jsem tuto vysokoškolskou kvalifikační práci vypracoval samostatně pod vedením vedoucího diplomové práce, s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 29. 5. 2011

.....

Poděkování:

Děkuji vedoucímu diplomové práce Ing. Martinovi Adámkovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování diplomové práce. Dále děkuji všem, kteří se podíleli na výrobě konstrukčních prvků a umožnili tak sestavení výrobku.

OBSAH

SEZNAM OBRÁZKŮ	11
ÚVOD.....	13
1 SENZOROVÁ TECHNIKA A ELEKTROCHEMICKÉ METODY	14
1.1 DEFINICE SENZORU.....	14
1.2 GENERACE SENZORŮ	14
1.3 OBECNÉ ROZDĚLENÍ SENZORŮ	14
1.4 ELEKTROCHEMICKÉ METODY	15
1.5 ELEKTROCHEMICKÉ METODY S VNUCOVANÝM NAPĚTÍM	15
1.6 ELEKTROCHEMICKÉ METODY S VNUCOVANÝM PROUDEM	17
1.7 TLV SENZORY	18
1.8 STÁVAJÍCÍ POSTUP MĚŘENÍ S ELEKTROCHEMICKÝMI SENZORY	19
2 VYBRANÉ PROGRAMOVACÍ JAZYKY.....	20
2.1 APACHE HTTP SERVER	21
2.2 MYSQL SERVER	21
2.3 HTML	21
2.4 JAVASCRIPT.....	23
2.5 PHP.....	23
2.6 C#	24
2.7 C++	25
2.8 C A JSA	26
3 NÁVRH AUTOMATIZACE MĚŘENÍ S ELEKTROCHEMICKÝMI SENZORY	27
3.1 SLED OPERACÍ PRO PŘÍPRAVU MĚŘENÉHO ROZTOKU.....	27
3.2 KONSTRUKČNÍ NÁVRH DÁVKOVACÍHO ZAŘÍZENÍ	27
3.3 KONSTRUKCE BĚŽCE.....	28
3.4 KONSTRUKCE STOJANU	30
4 REALIZACE ZAŘÍZENÍ PRO MÍCHÁNÍ ROZTOKŮ	31
4.1 PLASTOVÝ BĚŽEC	31
4.2 STOJAN	32
4.3 MANIPULACE S ROZTOKY.....	32
4.4 ZPĚTNÁ VAZBA ČERPÁNÍ	36
5 VÝBĚR HARDWAROVÉHO VYBAVENÍ.....	37
5.1 VÝBĚR MIKROKONTROLÉRU	37
5.1.1 Základní informace o ATmega	38
5.1.2 Architektura AVR mikrokotrolérů	38
6 VOLBA KOMUNIKAČNÍHO ROZHRANNÍ S PŘÍPRAVKEM.....	40
6.1 PROGRAMOVÁNÍ MIKROKONTORLÉRU ATMEGA PŘES USB	40
6.2 RS232-USB POMOCÍ FT232BM.....	41
6.3 ISP PROGRAMÁTOR	41
6.4 BOOTLOADER.....	43
6.4.1 Bootloader s hardwarovou inicializací	43
6.4.2 Bootloader kompatibilní s STK500	44

7	REALIZACE OBVODOVÝCH BLOKŮ	45
7.1	SBĚRNICE PRO MIKROKONTROLÉR ATMEGA128	45
7.2	ŘÍDICÍ JEDNOTKA	47
7.2.1	<i>Ovládání krokových motorů.....</i>	<i>48</i>
7.2.2	<i>Ovládání míchadla roztoků.....</i>	<i>48</i>
7.2.3	<i>Ovládání pump.....</i>	<i>49</i>
7.2.4	<i>Blok komunikace se senzory pro kontrolu čerpání</i>	<i>51</i>
7.3	JEDNOTKA POTENCIOSTATU.....	51
7.4	ČELNÍ PANEL	53
8	REALIZACE SOFTWARE VYBAVENÍ	55
8.1	NÁVRH VZDÁLENÉ KOMUNIKACE	55
8.2	EMBEDDED SOFTWARE.....	56
8.2.1	<i>Míchání roztoků.....</i>	<i>56</i>
8.2.2	<i>Měření s elektrochemickými roztoky.....</i>	<i>57</i>
8.2.3	<i>Kalibrace čerpání.....</i>	<i>57</i>
8.2.4	<i>Stavový automat.....</i>	<i>57</i>
8.3	OVLADAČ PRO OBSLUŽNÝ POČÍTAČ	58
8.3.1	<i>PC interface a LAN interface</i>	<i>59</i>
8.3.2	<i>Míchání roztoků.....</i>	<i>60</i>
8.3.3	<i>Měření se senzorem.....</i>	<i>61</i>
8.3.4	<i>Kamerový systém.....</i>	<i>62</i>
8.3.5	<i>LAN/WAN station.....</i>	<i>62</i>
8.4	INTERNETOVÁ APLIKACE	63
9	MĚŘENÍ	66
10	ZÁVĚR.....	67
	SEZNAM POUŽITÝCH ZDROJŮ.....	69
	PŘÍLOHY	1
	PŘÍLOHA 1 – PŘILOŽENÉ CD	1
	PŘÍLOHA 2 – SESTAVA BĚŽCE.....	2
	PŘÍLOHA 3 – KUSOVNÍKY PRO VÝROBU BĚŽCE	3
	PŘÍLOHA 4 – SESTAVA TĚLA ZAŘÍZENÍ	7
	PŘÍLOHA 5 – KUSOVNÍKY PRO VÝROBU TĚLA ZAŘÍZENÍ	8

Seznam obrázků

Obr. 1: Chronoamperometrie [7].	15
Obr. 2: Potenciodynamická metoda [7].	15
Obr. 3: Cyklická voltametrie [7].	16
Obr. 4: A.C. polarografie [7].	16
Obr. 5: Pulzní polarografie (Normal pulse polarography) [7].	16
Obr. 6: Diferenciální pulzní polarografie (Differential pulse polarography) [7].	16
Obr. 7: Schodovitá voltametrie (Staircase voltametry) [7].	17
Obr. 8: „Square-wave“ voltametrie [7].	17
Obr. 9: Galvanostatická chronopotenciometrie [7].	17
Obr. 10: Lineární (dynamická) chronopotenciometrie [7].	18
Obr. 11: Oscilografická polarografie [7].	18
Obr. 12: A.C. chronopotenciometrie [7].	18
Obr. 13: Schematické znázornění senzoru.	19
Obr. 14: Reálná podoba senzoru [7].	19
Obr. 15: Schéma měřicí aparatury [9].	19
Obr. 16: 3-D model běžce – pohled předo-horní.	29
Obr. 17: 3-D model běžce – pohled spodní.	29
Obr. 18: 3-D model běžce – pohled zado-horní.	29
Obr. 19: 3-D model celého zařízení.	30
Obr. 20: Realizace zařízení pro míchání roztoků.	31
Obr. 21: Plastový běžec – pohled shora.	32
Obr. 22: Plastový běžec - pohled ze spodu.	32
Obr. 23: Detail zařízení – umístění elektroniky.	32
Obr. 24: „Pomalá“ pumpa.	33
Obr. 25: „Rychlá“ pumpa.	33
Obr. 26: Návrh dvojramenného ventilu.	34
Obr. 27: Teorie tří expanzních nádob.	35
Obr. 28: Realizace expanzních nádob.	35
Obr. 29: Návrh přečerpací spojky.	35
Obr. 30: Součásti přečerpávací spojky.	36
Obr. 31: Složená přečerpávací spojka.	36
Obr. 32: Návrh optické průchodky.	36
Obr. 33: Realizace páru průchodek.	36
Obr. 34: Zapojení FT232BM.	41
Obr. 35: Zapojení ISP programátoru.	42
Obr. 36: Externí ISP programátor s redukcí.	42
Obr. 37: Fuse bity pro ATmega128 s bootloaderem.	43
Obr. 38: Schéma zapojení pro bootloader s hardwarovou inicializací.	44
Obr. 39: Externí zdroj pro podporu +5 V větve a zdroj +12 V větve.	45
Obr. 40: Zapojení čipu FTDI a rozvod napájecích napětí.	46
Obr. 41: Zapojení portů mikrokontroléru ATmega128.	46
Obr. 42: DPS redukce pro mikrokontrolér ATmega128.	47
Obr. 43: DPS sběrnice pro mikrokontrolér ATmega128.	47
Obr. 44: DPS řídicí jednotky.	47
Obr. 45: H-můstek pro ovládání cívky krokového motoru.	48
Obr. 46: Míchadlo roztoků.	48
Obr. 47: Zapojení bloku míchadla.	49
Obr. 48: Blok pro ovládání „pomalé“ pumpy.	50

Obr. 49: Blok pro ovládání „rychlé“ pumpy.....	50
Obr. 50: Blok komunikace se zpětnovazebními členy.	51
Obr. 51: DPS jednotky potenciostatu.	52
Obr. 52: Zapojení integrovaného obvodu IMAM [25].	52
Obr. 53: Obvodový blok jednotky potenciostatu.	53
Obr. 54: DPS čelního panelu.	54
Obr. 55: Detail zařízení – čelní panel.	54
Obr. 56: Konektory pro připojení senzorů.	54
Obr. 57: Návrh vzdálené komunikace se zařízením.	55
Obr. 58: Ovladač – výběr pracovního režimu.	58
Obr. 59: Ovladač – míchání roztoků.	60
Obr. 60: Ovladač – měření senzoru.	61
Obr. 61: Ovladač – LAN/WAN station.	62
Obr. 62: Internetová aplikace – přihlášení.	64
Obr. 63: Internetová aplikace – uživatelské prostředí.	65
Obr. 64: V/A charakteristika kontrolního roztoku.	66

Úvod

V dnešní době se schopnost komunikace přes Internet stala nejen jednou ze základních podmínek pro uspokojení stále vyšších nároků na flexibilitu nejrůznějších aplikací, ale i běžných každodenních potřeb. Internet se od doby svého vzniku stal důležitým nástrojem, který zasáhl téměř do všech oblastí počínaje komunikací a získáváním informací, až po internetové obchody, vzdálené bankovníctví, úřední správu a v neposlední řadě i školství [1].

Ve školství se rozšiřuje snaha o zakládání tzv. dálkově řízených laboratoří. Jedná se o automatizovaná pracoviště pro měření nejrůznějších elektrických i neelektrických veličin. Přínosem těchto laboratoří je možnost pohodlné kontroly dlouhodobých měření z libovolného místa s připojením na Internet, ale také zvýšení kvality výuky a zpřístupnění zajímavých vědeckých prací široké veřejnosti. Jedním z předních softwarů, které umožňují vzdáleně řídit procesy je LabView [2]. I přestože se jedná o nejrozšířenější formu, má řadu omezení a z pohledu uživatele internetového rozhraní není přívětivý, protože vyžaduje instalaci potřebného softwaru na PC uživatele. Jiný přístup byl zvolen na Matematicko-fyzikální fakultě UK v Praze, kde se vyvíjí tzv. ISES WEB Control [3]. Jedná se o typický příklad vzdáleně řízené laboratoře. I v tomto případě lze nalézt jisté nevýhody, a to hlavně v nutnosti instalace serveru. Na Internetu lze najít i mnoho dalších zajímavých projektů, které sice neřeší přímo provoz laboratoří, ale zabývají se vzdáleným řízením nejrůznějších zařízení, hraček, robotů, atd. Téměř u všech projektů je autentičnost podpořena videopřenosem z WWW kamery a okamžitými výsledky měření [4].

Předkládaná práce navazuje na dosažené výsledky v bakalářské práci autora [5], kde byla vytvořena plnohodnotná dálkově řízená laboratoř se zcela odlišným provedením komunikace než je tomu u ostatních obdobných dálkově řízených laboratoří. Přínosné výsledky a výstupy získané z bakalářské práce byly hybnou silou pro pokračování vývoje dálkově řízených zařízení.

Diplomová práce se zaměřuje na vývoj dálkově řízeného zařízení, pomocí něhož je možné realizovat automatizované pracoviště pro voltametrická měření v roztocích s tlustovrstvými elektrochemickými senzory. Jsou popsány jak vylepšení oproti původnímu řešení v bakalářské práci, tak i zcela nové poznatky hlavně z oblasti měření s elektrochemickými roztoky. Jsou zmíněny úpravy v hardwaru i softwaru a realizace nového zařízení.

1 Senzorová technika a elektrochemické metody

Hlavním cílem práce je navrhnout zařízení, které je schopné automaticky provést elektrochemická měření s tlustovrstvými elektrochemickými senzory. Pro návrh a realizaci je nutné znát základní informace o senzorech a vhodných metodách pro měření. Na základě získaných informací lze vytvořit základní myšlenku automatizace a návrh uživatelského prostředí, které bude pro budoucí uživatele srozumitelné.

1.1 Definice senzoru

Senzor je zařízení snímající sledovanou veličinu, kterou určitým principem transformuje na výstupní, lépe měřitelnou veličinu [6]. Skládá se z:

- citlivé části (snímací převodník),
- vyhodnocovacího obvodu (zesílení a vyhodnocení signálu, mikroprocesorové řízení, autokalibrace, ...).

1.2 Generace senzorů

Ve vývoji senzorové techniky byly zaznamenány jisté technologické mezníky, které určují čtyři generace senzorů [6]:

- první generace – jsou využívány makroskopické principy elektro-mechanické, elektrochemické nebo mechanické,
- druhá generace – jsou vyvíjeny jednočipové inteligentní senzory, označované jako SMART převodníky/senzory,
- třetí generace – neelektrická veličina působí na svazek světelného záření,
- MEMS – jsou založeny na mikroeletromechanických systémech umožňujících miniaturizaci druhé a třetí generace.

1.3 Obecné rozdělení senzorů

Mezi základní rozdělení senzorů patří dělení podle [6]:

- Druhu vstupní veličiny – geometrické, mechanické, teplotní, elektrické, magnetické, intenzita vyzařování, chemické, biologické, atd.
- Druhu výstupní veličiny – signál elektrický, optický, mechanický, dále analogová nebo digitální výstupní veličina.
- Styku senzoru s měřeným prostředím – dotykové a bezdotykové.
- Principu převodu – fyzikální, chemický, biochemický.
- Chování výstupu – aktivní (generátorové), pasivní.

- Převodu neelektrické veličiny – senzor s jednoduchým převodem, nebo s několikanásobným převodem.
- Výrobní technologie – senzory elektromechanické, mechanické, pneumatické, elektrické, elektronické, mikroelektronické, elektrochemické, optoelektronické apod.

1.4 Elektrochemické metody

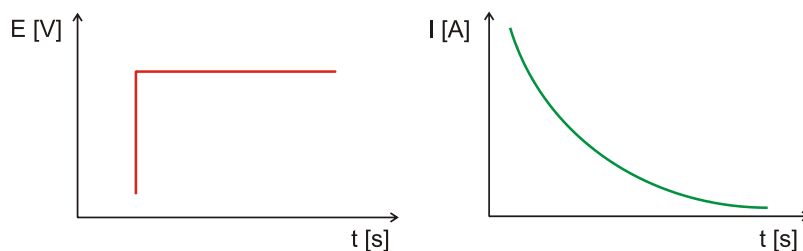
Elektrochemické metody [7] sledují chování pracovního prostředí v závislosti na jeho složení a koncentraci. Elektrochemické analytické metody v roztocích dělíme na:

- metody založené na elektrických vlastnostech roztoků – konduktometrie (měření vodivosti), dielektrimetrie (měření kapacity),
- metody s nulovým elektrolytickým proudem – potenciometrie,
- metody s nenulovým elektrolytickým proudem – voltametrie (polarografie), amperometrie, elektrogravimetrie, coulometrie.

1.5 Elektrochemické metody s vnucovaným napětím

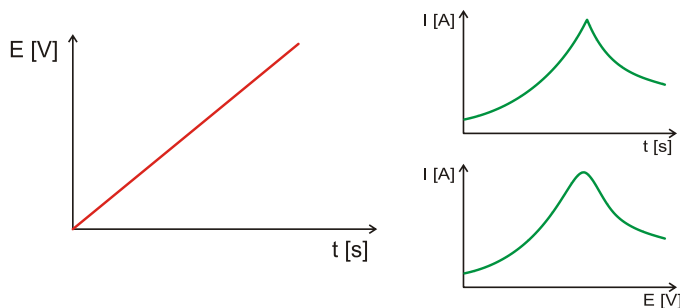
Metody jsou charakteristické tím, že je na elektrody připojeno definované napětí (potenciál) a je zaznamenáván proud procházející pracovní elektrodou, přičemž dle sledované veličiny je lze dělit na:

- sledování proudové odezvy systému na skokovou změnu potenciálu – chronoamperometrie, obr. 1,



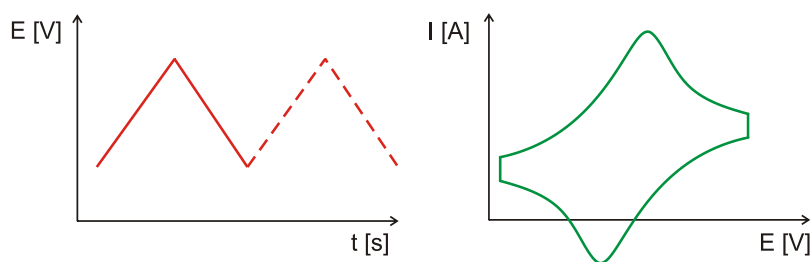
Obr. 1: Chronoamperometrie [7].

- sledování proudové odezvy systému na lineární změnu potenciálu – potenciodynamická metoda (stejnoseměrná voltametrie), obr. 2,



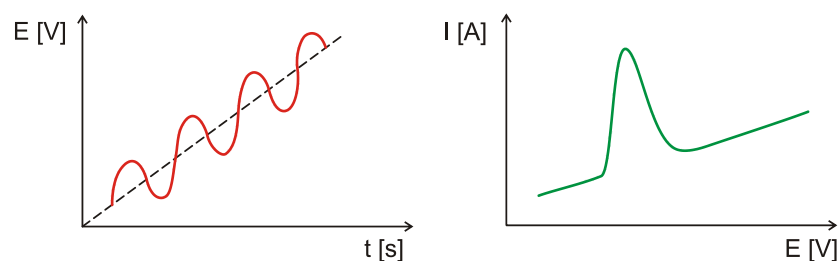
Obr. 2: Potenciodynamická metoda [7].

- sledování proudové odezvy systému na periodické opakování lineárního zvětšování a zmenšování potenciálu – cyklická voltametrie, obr. 3,



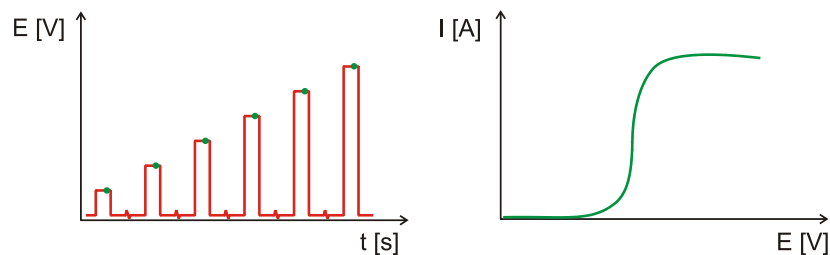
Obr. 3: Cyklická voltametrie [7].

- sledování proudové odezvy systému na lineární změnu potenciálu, na kterém je superponováno střídavé harmonické napětí s malou amplitudou – A.C. polarografie, obr. 4,



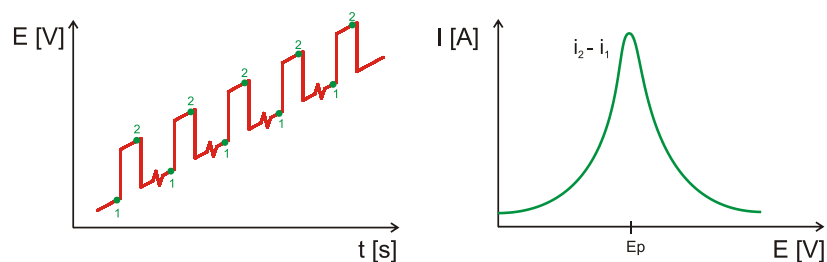
Obr. 4: A.C. polarografie [7].

- sledování proudové odezvy systému na obdélníkové impulzy s lineárně rostoucí amplitudou (proud je měřen v zeleně označených bodech průběhu) – pulzní polarografie, obr. 5,



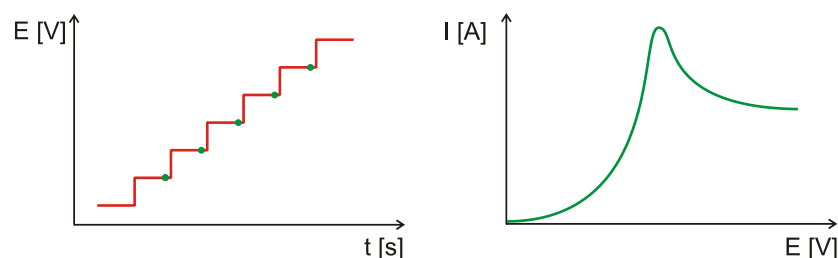
Obr. 5: Pulzní polarografie (Normal pulse polarography) [7].

- sledování proudové odezvy systému na obdélníkové impulzy superponované na lineárně rostoucí potenciál (proud je měřen v zeleně označených bodech průběhu) – diferenciální pulzní polarografie, obr. 6,



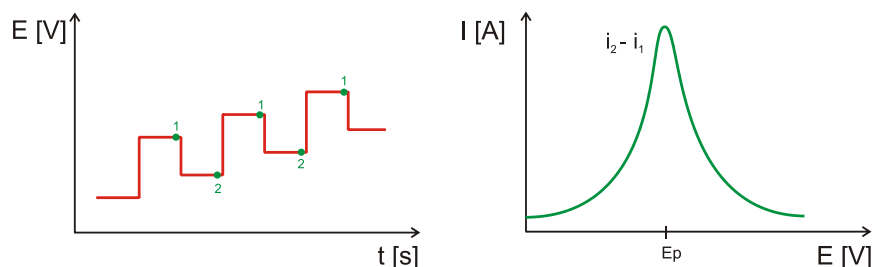
Obr. 6: Diferenciální pulzní polarografie (Differential pulse polarography) [7].

- sledování proudové odezvy systému na schodovitě rostoucí potenciál (proud je měřen v zeleně označených bodech) – schodovitá voltmetrie, obr. 7,



Obr. 7: Schodovitá voltmetrie (Staircase voltametry) [7].

- sledování proudové odezvy na lineárně rostoucí obdélkový signál se stálou amplitudou (proud je měřen v zeleně označených bodech průběhu) – „square-wave voltametrie“, obr. 8,

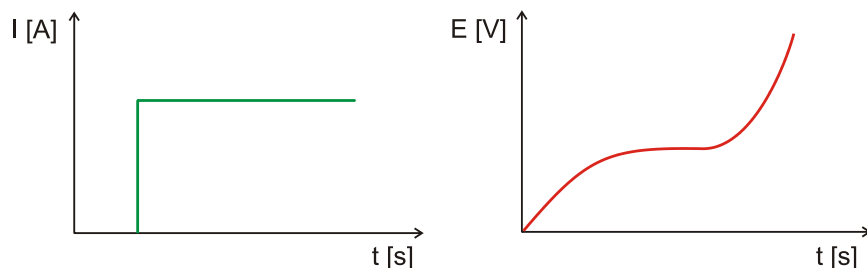


Obr. 8: „Square-wave“ voltametrie [7].

1.6 Elektrochemické metody s vnucovaným proudem

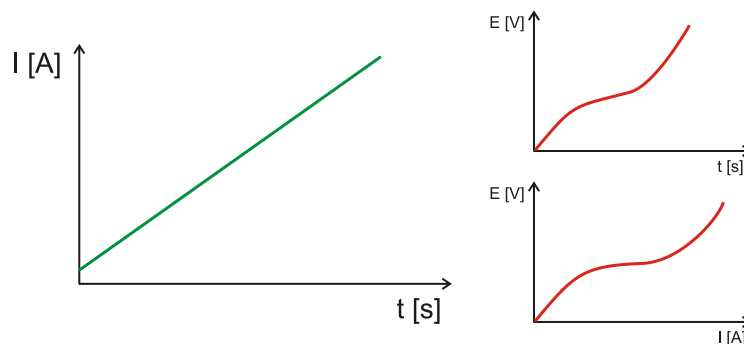
Metody jsou charakteristické tím, že elektrodami prochází definovaný proud a je zaznamenáváno napětí (potenciál) mezi pracovní a referenční elektrodou, přičemž dle sledované veličiny je lze dělit na:

- sledování potenciálové odezvy na skokovou změnu proudu – galvanostatická chronopotenciometrie, obr. 9,



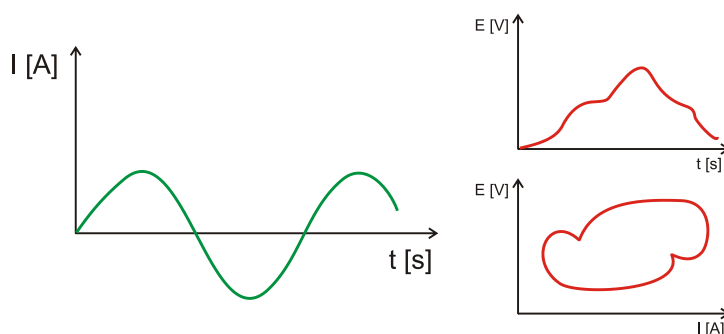
Obr. 9: Galvanostatická chronopotenciometrie [7].

- sledování potenciálové odezvy na lineární změnu proudu – lineární (dynamická) chronopotenciometrie, obr. 10,



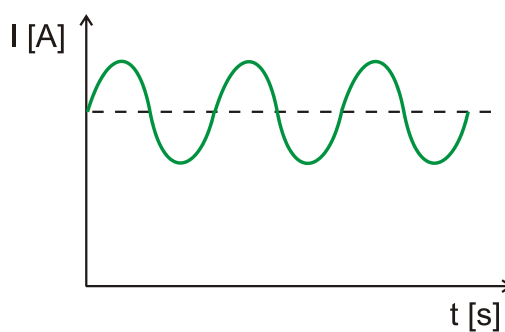
Obr. 10: Lineární (dynamická) chronopotenciometrie [7].

- sledování potenciálové odezvy na periodickou změnu proudu – oscilografická polarografie, obr. 11,



Obr. 11: Oscilografická polarografie [7].

- sledování odezvy systému na střídavý harmonický signál s malou amplitudou a konstantním offsetem – A.C. chronopotenciometrie, obr. 12.



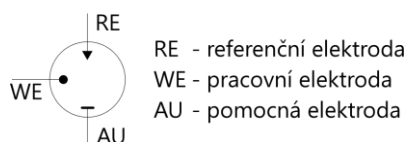
Obr. 12: A.C. chronopotenciometrie [7].

1.7 TLV senzory

Zařízení je navrhováno pro měření se senzory realizovanými pomocí technologie tlustých vrstev [8]. Technologie spočívá v nevakuovém nanášení funkčních vrstev na keramický nosný substrát. Vrstvy jsou nanášeny v podobě pasty formou sítotisku nebo šablonového tisku. Operace nanášení mohou být odděleny procesem levelingu (ustálení nanesené vrstvy) a vysušením při 70° - 150° C po dobu 15 až 30 minut. Po nanesení určitého počtu požadovaných vrstev následuje vysokoteplotním nevakuový výpal při teplotě 850° C po dobu 10 minut s hodinovým teplotním profilem.

Pasty pro technologii tlustých vrstev jsou složeny ze tří složek. Jsou to složka funkční, tavivová a pojivová. Funkční složka určuje elektrické vlastnosti pasty. Tavivová složka v sobě váže částice funkční složky a je tvořena skleněnou fritou. Pojivová složka zajišťuje tiskové vlastnosti pasty, především její viskozitu.

Použité senzory lze zařadit do kategorie pevných senzorů. Obsahují tři elektrody – pracovní, referenční a pomocnou. Schematické znázornění senzoru je na obr. 13 a jeho reálná podoba je zobrazena na obr. 14.



Obr. 13: Schematické znázornění senzoru.

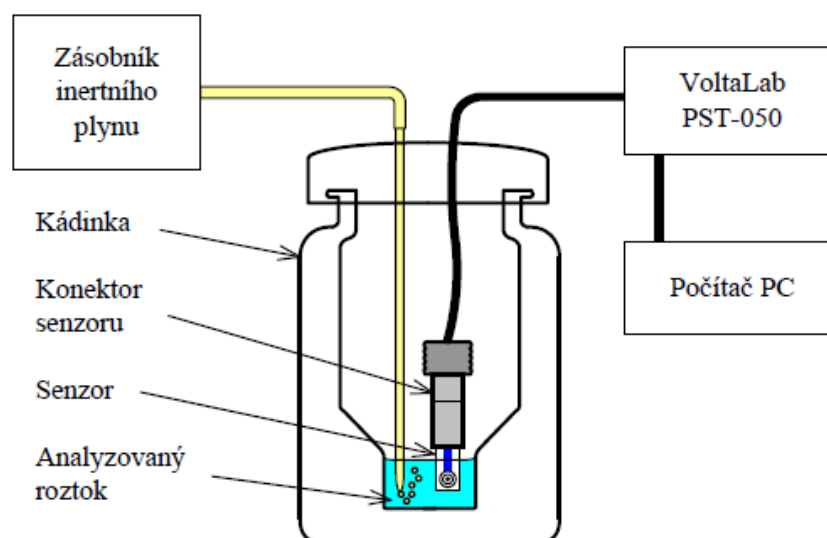
Obr. 14: Reálná podoba senzoru [7].

1.8 Stávající postup měření s elektrochemickými senzory

Základem měřicí aparatury [9] v laboratoři Ústavu mikro-elektroniky, FEKT, VUT v Brně, pro kterou byl přístroj vyvíjen, je reakční nádoba (obr. 15). V ní je umístěn měřený roztok, do něhož je vsunut senzor a dále je možno přivést ochrannou dusíkovou atmosféru z tlakové nádoby.

Výstupní proudová odezva senzoru je měřena pomocí analytické elektrochemické pracovní stanice VoltaLab PST050, Radiometer Analytical. Jedná se o potenciostat (zařízení pro měření proudu při definovaném průběhu potenciálu), který umožňuje měření různých elektrochemických metod – otevřený obvod, metoda potenciostatická, galvanostatická, potenciodynamická, galvanodynamická,...

Vyhodnocení měření je prováděno počítačem pomocí programů „VoltaMaster 4“ a „Microsoft Excel“.



Obr. 15: Schéma měřicí aparatury [9].

2 Vybrané programovací jazyky

V oblasti programování softwaru a internetových aplikací převládají vždy určité trendy ve výběru programovacích jazyků. Je vhodné vybírat programovací jazyky, s nimiž pracují velké skupiny programátorů a rozhodnout se pro ten, který je pro danou aplikaci nejvhodnější.

U internetových aplikací je navíc nutné uvažovat o tom, zda je možné financovat server s podporou vybraného jazyka. Tento aspekt hraje ve výběru značnou roli, a proto v dnešní době Internet silně ovládá tzv. „Webserver triáda“ – server Apache + databázový server MySQL + programovací jazyk PHP. Navzdory tlaku softwarového nástroje .NET, který je v určitých oblastech mnohem silnější a má výborné vývojové prostředí, je tato zmíněná triáda využívána velkou částí programátorů, protože je zcela zdarma.

Pro programování pozadí dynamických internetových stránek a uživatelských prostředí je možné volit hlavně mezi PHP, ASP .NET a Javou. Pro tvorbu vzhledu a vytváření formulářových polí (interface) je využíván jazyk HTML, nebo XHTML, CSS a JavaScript, případně AJAX. Přičemž JavaScript a AJAX je na pomezí mezi programovým pozadím a interfacem internetových stránek a vytváří jakousi vazbu mezi oběma oblastmi. Právě tyto jazyky umožňují vytvářet velice dynamicky se chovající internetové stránky.

Ve výběru programovacího jazyka pro vývoj formulářových aplikací pro Windows je mnohem větší volnost, ale jako nejvhodnější se jeví C#. Důvodem je, že se jedná o programovací jazyk vyvíjený přímo společností Microsoft a obsahuje řadu potřebných knihoven a mimo jiné i obsáhlou dokumentaci.

Software pro mikrokontroléry lze vyvíjet přímo v JSA (jazyk symbolických adres), ovšem s výhodou se dají použít i vyšší programovací jazyky a kompilátory vyvíjené přímo výrobcí mikročipů. Mezi běžně používané vyšší jazyky patří C a C++, jejichž kompilací lze efektivně dosáhnout požadovaného cíle bez nutnosti zásahu do JSA. Rychlosti a efektivity programu vytvořeného přímo v JSA kompilátory sice nedosáhnou, ale tvorba tzv. embedded software (zabudované programové vybavení) pomocí vyšších programovacích jazyků přináší řadu výhod:

- tvorba programů v týmu (objektové programování),
- přehlednost vytvořeného programu,
- rychlý vývoj a úprava,
- značná podpora ze strany programátorů nejen z oblasti programování mikrokontrolérů, atd.

2.1 Apache HTTP server

Apache HTTP Server [10] je v současné době nejrozšířenějším webovým serverem, který umožňuje provoz internetových stránek. Je vyvíjen jako Open source aplikace pro Linux, BSD, Microsoft Windows a další platformy.

Vývoj Apache začal v roce 1993 v NCSA (National Center for Supercomputing Applications) na Illinoiské univerzitě. Původní jméno projektu bylo NCSA HTTPd. V dalším roce však vývojářský tým opustil hlavní programátor Rob McCool, tím došlo ke zpomalení vývoje a poté, v roce 1998, k úplnému zastavení.

NCSA HTTPd však mezitím už používali správci webových serverů a dodávali k němu vlastní úpravy – „patche“. Hlavní úlohu v dalším vývoji sehráli Brian Behlendorf a Cliff Skolnick, kteří začali koordinovat sběr úprav a jejich distribuci.

První veřejná verze s označením 0.6.2 byla vydána v dubnu 1995. Následovalo kompletní přepsání kódu a založení Apache Group. Od dubna 1996 byl Apache nejpopulárnější server na Internetu. V květnu 1999 běžel na 57 % všech serverů a v listopadu 2005 jeho používanost dosáhla 69 %.

2.2 MySQL server

MySQL [11] je relační databázový systém typu DBMS (database management system). Každá databáze v MySQL je tvořena z jedné nebo více tabulek, které mají řádky a sloupce. V řádcích rozeznáváme jednotlivé záznamy jako hodnoty polí jednotlivých sloupců určených názvy. Sloupce uvozují datový typ záznamu. Práce s databázemi, tabulkami a daty se provádí pomocí příkazů, respektive dotazů. Dotazy vycházejí z deklarativního programovacího jazyka SQL (Structured Query Language). Systém MySQL je využitelný v C, C++, Java, Perl, PHP, Python, Tcl, Visual Basic, .NET a je šířen jako Open source .

MySQL bylo od počátku optimalizováno především na rychlost, z čehož plynou i určitá omezení. Má jen jednoduché způsoby zálohování, a až donedávna nepodporovalo pohledy, triggerly a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech, protože začaly být vyžadovány uživateli hlavně ze strany programátorů webových stránek.

2.3 HTML

HTML [12] nebo-li HyperText Markup Language, je značkovací jazyk a využívá se k vytváření internetových stránek. Byl vyvinut z rozsáhlého univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language), ale projevila se i vazba na vývoj webových prohlížečů, který zpětně ovlivňoval definici jazyka.

V roce 1989 spolupracovali Tim Berners-Lee a Robert Caillau na propojeném informačním systému pro CERN. V té době se pro tvorbu dokumentů obvykle používaly jazyky TeX, PostScript a také SGML. Berners-Lee si uvědomoval potřebu jednoduššího jazyka a v roce 1990 bylo proto navrženo HTML a protokol pro jeho přenos v počítačové síti – HTTP (HyperText Transfer Protocol). Zároveň byl napsán první webový prohlížeč nazvaný WorldWideWeb.

Jazyk HTML je charakterizován množinou značek a jejich atributů definovaných pro danou verzi. Názvy jednotlivých značek se uzavírají mezi úhlové závorky <, >. Sémantika nebo-li význam části textu dokumentu se určuje uzavřením textu mezi určité značky a vytváří se tzv. element (prvek) dokumentu. Každý element může být doplněn atributy, které upřesňují vlastnosti. Značky (zvané tagy) jsou obvykle párové, přičemž koncová značka je shodná se značkou počáteční, jen má před názvem znak lomítko. Příklad pro označení odstavce:

```
<p>Text odstavce</p>.
```

Některé značky jsou nepárové – nemají žádný obsah a nepoužívají koncovou značku. Příklad pro vykreslení vodorovné čáry:

```
<hr>.
```

Příklad na atributy elementu se dají názorně ukázat na různém zápisu odkazu:

- základní typ zápisu `<a href="http://example.com">text odkazu</a>`,
- odkaz, který se otevře v novém okně/panelu `<a href=http://example.com target="_blank">text odkazu</a>`,
- odkaz, který se otevře v novém okně/panelu, obsahuje titulek (zobrazí se po najetí myší) a bude přiřazen ke třídě CSS `<a href="http://example.com" target="_blank" title="titulek" class="navez_tridy">text odkazu</a>`.

Dokument v jazyku HTML má předepsanou strukturu:

- deklarace DTD – je povinná až ve verzi 4.01, je uvedena direktivou, která se liší jednotlivými verzemi HTML, např. `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">`,
- kořenový element – značky `<html> a </html>`, reprezentuje celý dokument,
- hlavička elementu – značky `<head> a </head>`, obsahuje metadata, která se vztahují k celému dokumentu, definují např. název dokumentu, jazyk, kódování, klíčová slova, popis, použitý styl zobrazení,
- tělo dokumentu – značky `<body> a </body>`, obsahuje vlastní text dokumentu.

Elementy html, head a body jsou povinné, ale jejich otevírací a koncové značky povinné nejsou, prohlížeč si je sám doplní podle kontextu.

2.4 JavaScript

JavaScript [13] je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich. Nyní se zpravidla používá jako interpretovaný programovací jazyk pro WWW stránky, často vkládaný přímo do HTML kódu stránky. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků.

Jeho syntaxe patří do rodiny jazyků C/C++/Java. Slovo Java je však součástí jeho názvu pouze z marketingových důvodů a s programovacím jazykem Java jej vedle názvu spojuje jen podobná syntaxe. JavaScript byl v červenci 1997 standardizován asociací ECMA (European Computer Manufacturers Association) a v srpnu 1998 ISO (International Organization for Standardization). Standardizovaná verze JavaScriptu je pojmenována jako ECMAScript a z ní byly odvozeny i další implementace, jako je například ActionScript.

JavaScript byl původně obchodní název implementace společnosti Netscape, kde byl vyvíjen nejprve pod názvem Mocha, později LiveScript, ohlášen byl společně se společností Sun Microsystems v prosinci 1995 jako doplněk k jazykům HTML a Java. Pro verzi firmy Microsoft je použit název JScript. JScript je podporován platformou .NET. Program v JavaScriptu se obvykle spouští až po stažení WWW stránky z Internetu (tzv. na straně klienta), na rozdíl od ostatních jiných interpretovaných programovacích jazyků (např. PHP a ASP), které se spouštějí na straně serveru ještě před stažením z Internetu. Z toho plynou jistá bezpečnostní omezení, JavaScript např. nemůže pracovat se soubory, aby tím neohrozil soukromí uživatele.

2.5 PHP

PHP (Hypertext Preprocessor) [14] je v současnosti velmi rozšířená technologie umožňující snadné programování na straně serveru a využívá se k tvorbě různých interaktivních webových stránek. Skript je proveden na serveru podle zadaných kritérií a výsledek je odeslán volajícímu počítači stejným způsobem, jakým se odesílají běžné statické (HTML) stránky. Jakmile je však stránka načtena klientem, pomocí PHP ji již není možné dále měnit. Změnu stránky bez nutnosti jejího opětovného načtení se provádí například pomocí JavaScriptu. Výhoda tohoto programovacího jazyka spočívá v tom, že není třeba pro změnu stránky neustále obnovovat obsah stránky. Nevýhodou je nízká bezpečnost jazyka. Z tohoto důvodu se JavaScript využívá spíše pro okamžitou úpravu CSS kódu, či jako doplněk při vytváření webové grafiky.

PHP [15] je programovací jazyk umožňující procedurální i objektově orientované programování. Znalost objektově orientovaného programování (OOP) může být při práci v PHP výhodou, není však nutnou podmínkou. PHP také patří mezi jazyky, kde například není nutné předem definovat typ proměnných, navíc jakákoli proměnná může kdykoli změnit svůj typ. Webová stránka s prvky PHP má nejčastěji koncovku php avšak je možné použít i php3, php4, php5 nebo phtml.

2.6 C#

C# [16] je vysoce úrovněový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft zároveň s platformou .NET Framework. Microsoft založil C# na jazycích C++ a Java (a je tedy nepřímým potomkem jazyka C, ze kterého čerpá syntaxi).

C# lze využít k tvorbě databázových programů, webových aplikací a stránek, webových služeb, formulářových aplikací ve Windows, softwaru pro mobilní zařízení atd.

Standard ECMA definuje současný design C# takto:

- C# je jednoduchý, moderní, mnohoúčelový a objektově orientovaný programovací jazyk.
- Jazyk a jeho implementace poskytuje podporu pro principy softwarového inženýrství jako jsou: hlídání hranic polí, detekce použití neinicializovaných proměnných a automatické uvolňování paměti pomocí funkce garbage collector. Důležité jsou také jeho vlastnosti jako: robustnost, trvanlivost a programátorská produktivita.
- Jazyk je vhodný pro vývoj softwarových komponent distribuovaných v různých prostředích.
- Přenositelnost zdrojového kódu je velmi důležitá, obzvláště pro ty programátory, kteří jsou obeznámeni s C a C++.
- C# je navržen pro psaní aplikací jak pro zařízení se sofistikovanými operačními systémy, tak pro zařízení s omezenými možnostmi.
- Přestože by programy psané v C# neměly plýtvat s přiděleným procesorovým časem a pamětí, nemohou se měřit s aplikacemi psanými v C nebo jazyce symbolických adres.

Jazyk C# je navržen tak, aby co nejvíce zohledňoval strukturu Common Language Infrastructure (CLI), se kterou je používán. Většina základních typů v C# přímo odpovídá základním typům v platformě CLI. Návrh jazyka, ale nevyžaduje, aby překladač generoval Common Intermediate Language (CIL) nebo jiný konkrétní formát.

Teoreticky je možné, aby překladač vytvářel strojový kód podobný běžným překladačům jazyka C++ a jiných, ale v praxi všechny překladače jazyka C# generují CLI.

2.7 C++

C++ [17] je objektově orientovaný programovací jazyk, který byl vyvinut v Bellových laboratořích AT&T pod vedením Bjarna Stroustrupa, rozšířením jazyka C. Není čistě objektovým jazykem, protože podporuje i procedurální a generické programování. Starší verze jazyka, společně označované jako „C with Classes“ (C s třídami), byly používány od roku 1980. Jméno „C++“ vymyslel Rick Mascitti a zdůrazňuje evoluční povahu změn oproti jazyku C.

První překladače C++ byly preprocesory, které překládaly z C++ do čistého C. C++, ale není pouhé rozšíření jazyka C, protože s ním není zcela kompatibilní. Koncepce objektů jazyka C++ byla převzata z jazyka Simula 67. Objekty (třídy) jsou pojaty jako přirozené rozšíření datových struktur jazyka C o možnost vkládání členských funkcí. C++ umožňuje řídit viditelnost složek objektů pro ostatní části programu. Pro objekty je možná vícenásobná dědičnost, která je hlavní výhodou objektově orientovaného programování. Hlavní myšlenka dědičnosti je „znuvupoužitelnost“, tedy možnost vytvářet nové třídy založené na třídě, která již byla definována. Díky dědičnosti je možné napsat kód jednou pro obecnější typ a poté ho používat pro všechny jeho potomky. Šablony dále rozšiřují „znuvupoužitelnost“ kódu, neboť umožňují napsat kód se zcela obecným (neurčeným) datovým typem. Jazyk C++ umožňuje deklarovat více funkcí se stejným názvem. Kompilátor určí správné použití podle počtu a typu parametrů. Tato technika se nazývá přetěžování funkcí. Velmi silnou vlastností jazyka je i možnost přetěžovat standardní operátory (například '+' nebo '=') a tak přirozeně využívat tyto operátory pro nově vytvářené třídy a tvorbu abstraktních datových typů.

Standard jazyka C++ z roku 1998 se skládá ze dvou částí - popis jazyka a standardní knihovny. Standardní knihovna jazyka C++ obsahuje mírně modifikovanou verzi standardní knihovny jazyka C a Standard Template Library (STL). STL obsahuje velké množství užitečných datových struktur a algoritmů, jako například vektory, spojové seznamy, iterátory, zobecněné ukazatele, (multi)mapy, (multi)sety. Všechny tyto struktury mají konzistentní rozhraní. S použitím šablon je pak možné programovat generické algoritmy schopné pracovat s kterýmkoliv kontejnerem nebo sekvencí definovanou iterátory. Většina kompilátorů poskytuje implementaci standardu C++ včetně STL, ale existují také implementace standardu nezávislé na kompilátoru nebo jiné projekty vytvářené pro různé zákaznické implementace s různými cíli návrhu.

2.8 C a JSA

JSA [18], nebo-li jazyk symbolických adres (anglicky assembly language), je nízkoúrovňový programovací jazyk, který je tvořen symbolickou reprezentací jednotlivých strojových instrukcí a konstant potřebných pro vytvoření strojového kódu pro daný procesor. Symbolickou reprezentaci tvoří zpravidla výrobce procesoru a je založena na mnemotechnických zkratkách. JSA je proto závislý na konkrétním procesoru a zapsaný program je obtížně přenositelný na jinou platformu. Pro překlad JSA do strojového kódu se používá program, který se nazývá assembler (překladač).

JSA se poprvé objevily v 50. letech 20. století, kdy byly označovány jako druhá generace programovacích jazyků. Eliminovaly většinu chyb vznikajících při programování a časovou náročnost první generace programovacích jazyků. Ve své době byly široce využívány pro všechny typy programování.

Pokročilé překladače JSA poskytují doplňující nástroje pro správu a vývoj kódu, řízení překladu programu, a podporu ladění. Což umožňuje využít vyšší programovací jazyky jako např. C pro programování procesorů. Problematický je odhad velikosti výsledného zdrojového kódu v JSA, protože vytváření instrukcí je závislé na nastavení, optimalizaci a dalších faktorech souvisejících přímo s konkrétním překladačem. I přesto se, ale pro programování mikrokontrolérů využívají vyšší programovací jazyky, protože značným způsobem šetří čas při vývoji a mnohdy se jimi dají řešit i úlohy v JSA extrémně obtížné.

3 Návrh automatizace měření s elektrochemickými senzory

Návrh zařízení, které bude schopné automatizovat elektrochemická měření, je založeno na dodržení sledu operací, které jsou nutné pro namíchání měřeného roztoku. Cílem automatizace je umožnit uživateli přesné namíchání roztoku bez nutnosti manuální manipulace s jednotlivými složkami. Důsledkem automatizace je možnost řídit proces míchání i vzdáleně s vyloučením přítomnosti obsluhy u zařízení. Tato kapitola se věnuje shrnutí základních parametrů pro přípravu měření a popisuje způsob návrhu jednotlivých konstrukčních prvků realizovaného zařízení.

3.1 *Sled operací pro přípravu měřeného roztoku*

Postup přípravy roztoku pro měření je následující:

1. čerpání pufru (základní roztok) do pipety (200 ml),
2. vypuštění pufru do reakční nádoby,
3. čerpání roztoku z ampule do dávkovače (25 μ l - 100 μ l) – může se opakovat i pro více roztoků a v průběhu měření je měřený roztok obohacován o stanovené dávky,
4. vypuštění roztoku do reakční nádoby.

Při přípravě roztoku je nutné zajistit jeho homogenitu pravidelným promícháváním. Po přípravě následuje měření s elektrochemickým senzorem. Pokud je měření ukončeno, musí se aparatura před zahájením nového měření vyčistit. Postup čištění je následující:

1. vyčerpání obsahu reakční nádoby,
2. vypuštění obsahu do odpadiště,
3. čerpání oplachu,
4. vypuštění oplachu do reakční nádoby,
5. promíchání obsahu reakční nádoby,
6. vyčerpání obsahu reakční nádoby,
7. vypuštění použitého oplachu do odpadiště.

3.2 *Konstrukční návrh dávkovacího zařízení*

Z dodržení sledu operací pro namíchání měřeného roztoku vychází několik podmínek, které musí navrhované zařízení splňovat:

- čerpání malých (v řádech μ l) i velkých (v řádech ml) objemových množství,

- schopnost čerpat více typů roztoků,
- schopnost vyčerpat obsah reakční nádoby,
- zajistit homogenitu namíchaného roztoku.

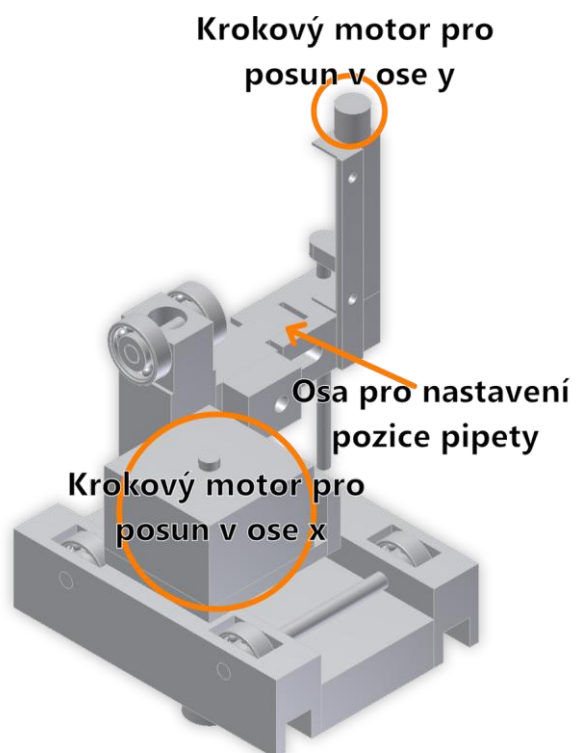
Navržené zařízení bylo převedeno do 3-D modelu, jehož součásti budou popsány v následujícím textu. Návrh splňuje veškeré podmínky stanovené ze sledu operací pro namíchání roztoku. V návrhu se neuvažuje ani konkrétní vzhled zařízení ani způsob realizace čerpání či míchání roztoku. Návrh slouží pouze pro hrubou představu možného řešení a hlavně pro vytvoření výkresů pro výrobu jednotlivých součástí. Výkresy s kusovníky a sestavami jsou v příloze a jejich následná úprava bude popsána v části věnující se praktické realizaci.

3.3 Konstrukce běžce

Běžec je určen pro přepravu roztoků z jednotlivých ampulí do reakční nádoby, ale také pro její naplnění pufrem, odčerpávání a oplachování jejího obsahu po provedení měření. Reakční nádoba, nebo taky „terč“, je shromažďovací kádinka o objemu cca 250 ml, v níž probíhá měření s elektrochemickými senzory. Obsahuje i systém pro promíchávání svého obsahu tak, aby byla zajištěna jeho homogenita. Promíchávání slouží i pro důkladný oplach.

Úkolem běžce je automatizace procesu měření. Tedy příprava roztoku pro měření a čištění terče. Běžec je umístěn ve stojanu tak, aby se za pomoci krokového motoru mohl pohybovat v jedné ose. Navíc je opatřen druhým krokovým motorem, který řídí chod přečerpávacího hrotu. Posuv hrotu v ose y je maximálně 5 cm. Oba krokové motory mají možnost kalibrace polohy. Krovový motor pro posuv v ose x lze i naklánět tak, aby bylo možné zvolit potřebný přítlak gumového kotouče na stěnu pojezdových lišt. Tělo běžce je opatřeno řadou kuličkových ložisek. Dva páry jsou umístěny tak, aby ulehčili chod po pojezdových lištách. Jeden pár je pružně umístěn na vrcholu těla a zajišťuje, aby se běžec při posuvu po stojanu nekýval nebo případně nesklouznul a také aby gumový kotouč měl konstantní přítlak na stěnu lišty.

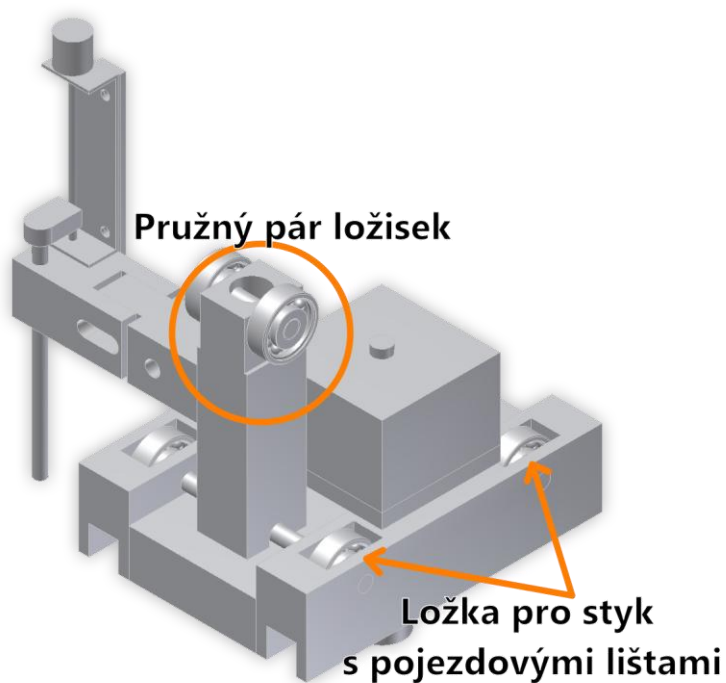
Uvedené 3-D modely na obr. 16 – obr. 18 jsou jen orientační, protože neobsahují detailně zpracované osy, šrouby a krokové motory. Je z nich ovšem patrná základní myšlenka konstrukčního návrhu běžce.



Obr. 16: 3-D model běžce – pohled předo-horní.



Obr. 17: 3-D model běžce – pohled spodní.



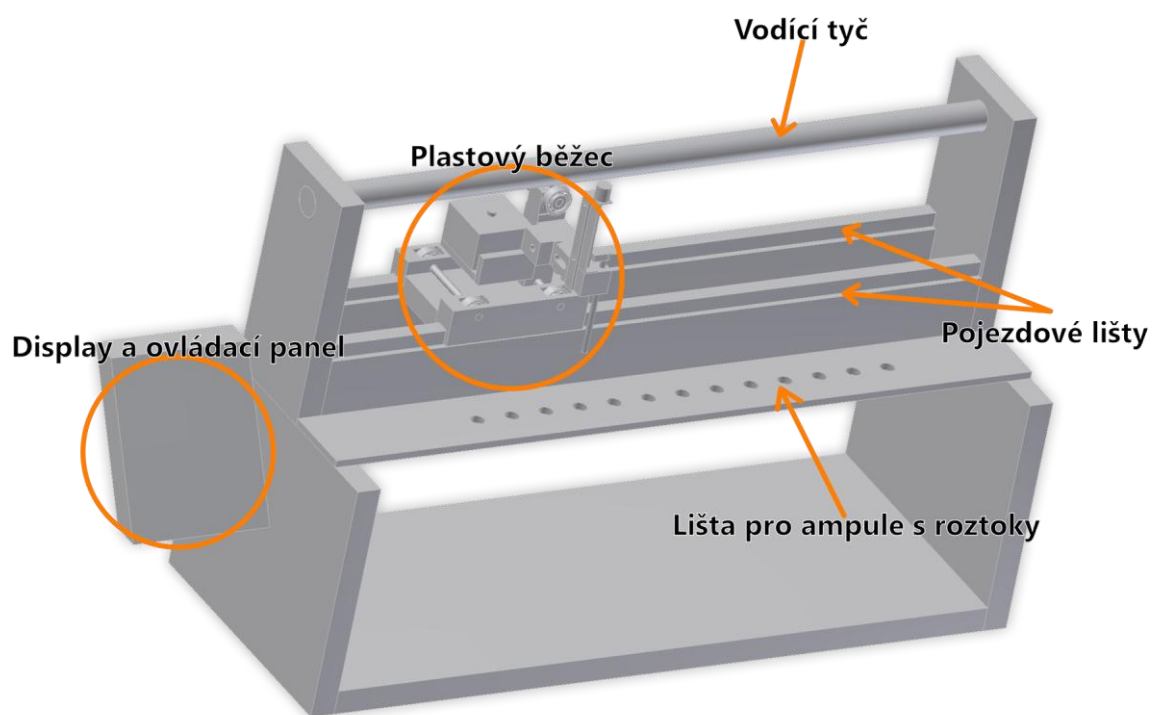
Obr. 18: 3-D model běžce – pohled zado-horní.

3.4 Konstrukce stojanu

Základní funkcí stojanu (obr. 19) je zajištění přesného pohybu běžce na definované polohy jednotlivých otvorů, v nichž jsou umístěny ampule s roztoky, přečerpávací spojky a vypouštěcí otvor pro reakční nádobu a odpadiště.

Konstrukce se skládá z:

- pojezdových lišt,
- vodící tyče,
- precizně rozmístěných otvorů pro ampule a ostatní funkční otvory,
- panel pro klávesnici a displej,
- prostor pro pumpy, kádinky, míchadlo a elektroniku.



Obr. 19: 3-D model celého zařízení.

4 Realizace zařízení pro míchání roztoků

Navržené 3-D modely a výkresy kusovníků a sestav sloužily jako předloha pro výrobu. Výroba byla kombinovaná a skládala se z obrábění na CNC stroji, CNC fréze a laserového řezání. Obráběcím strojem se vyrobil běžec z tvrdého průmyslového plastu a provedla se povrchová úprava jednotlivých součástí stojanu. Stojan byl vyroben z ocelových plátů řezáním laserem a za pomoci frézy bylo vyrobeno ozubení na jedné z pojezdových lišt. Pro přesné zasednutí do ozubení na pojezdové liště bylo vyrobeno i ozubené kolečko s modulem 18 pro krokový motor. Sestavení celého zařízení (obr. 20) bylo realizováno částečně obloukovým svařováním a částečně sešroubováním.

Od návrhu vycházejícího z 3-D modelů se upustilo hlavně v realizaci vodící tyče, která se v praxi neosvědčila a vedla spíše k problémům při posuvu běžce. Dále byl nahrazen gumový válec ozubeným kolečkem a ozubením na pojezdové liště a poslední zásadní odchylkou byla realizace obslužného panelu. Panel se přemístil do čela přístroje, protože byla odstraněna klávesnice pro ovládání a bylo nutné vytvořit místo pro kontakty stíněných vodičů vedoucích od senzoru. Celé zařízení bylo nalakováno bílou barvou a čelní panel byl opatřen grafikou pro větší srozumitelnost jednotlivých prvků.

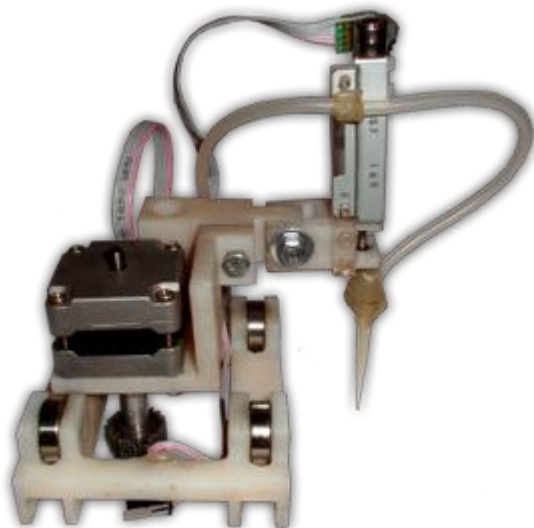


Obr. 20: Realizace zařízení pro míchání roztoků.

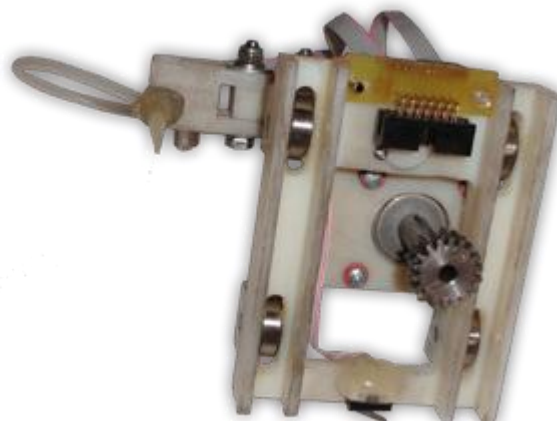
4.1 Plastový běžec

Na plastovém běžci (obr. 21 a 22) byly úpravy vedeny hlavně ve směru odlehčení celé konstrukce, a proto se odstranily nadbytečné plochy materiálu. Protože byla

odstraněna vodící tyč, byl odstraněn i pružný pár ložisek, které se o tuto tyč původně opíraly. Zbylý koncept běžce byl zachován a na jeho tělo byly instalovány krokové motory, DPS pro rozvod signálových a napájecích vodičů, vodící tyč, na jejímž konci byl upevněn přečerpávací hrot a spínače pro autokalibraci. Pro snadnost instalace je na DPS běžce kontakt pro plochý kabel, jímž je zprostředkována komunikace a v těle je mosazná průchodka pro propojení silikonové hadice.



Obr. 21: Plastový běžec – pohled shora.



Obr. 22: Plastový běžec - pohled ze spodu.

4.2 Stojan

V prázdném prostoru stojanu byla umístěna expanzní nádoba, míchadlo roztoků, reakční nádoba, nádoba s pufrem, nádoba s oplachem, nádoba pro odpad a pumpy. Do oblasti pod pojezdovými lištami byla upevněna elektronika viz. obr. 23.



Obr. 23: Detail zařízení – umístění elektroniky.

4.3 Manipulace s roztoky

Stojan a plastový běžec umožňují definovat přesné pozice pro konkrétní roztoky a funkční otvory. Běžec, proto může provádět posun na následující pozice:

- P1 - P8 – pozice obsahující roztoky v ampulích,
- pufr – pozice sloužící pro čerpání základního roztoku přes přečerpávací spojku,
- oplach – pozice sloužící pro čerpání oplachu přes přečerpávací spojku,
- senzor – pozice sloužící pro vypouštění roztoků do reakční nádoby skrze průchodku s optickými senzory zajišťujícími zpětnou vazbu pro vypouštění,
- odsávání – pozice sloužící pro čerpání obsahu reakční nádoby přes přečerpávací spojku,
- odpad – pozice sloužící pro vypouštění nepotřebných roztoků a produktů po čištění reakční nádoby skrze průchodku s optickými senzory zajišťujícími zpětnou vazbu pro vypouštění.

Manipulace s roztoky je zajištěna posunem běžce na definované pozice a jejich čerpání umožňují dvě pumpy. Potřeba dvou pump vychází z praktického hlediska nutnosti čerpání velkých i malých objemových množství. Protože peristaltickou („pomalou“) pumpou (obr. 24), která umí čerpat rychlostí v řádech μl za minutu, by bylo velmi časově náročné čerpat objemová množství, jako je např. 200 ml pufru, byla zařazena i druhá „rychlá“ pumpa (obr. 25). Tato pumpa není tak přesná, ale definované čerpané množství se u ní pohybuje v řádech ml za minutu.

Obě pumpy byly upraveny tak, aby se daly řídit digitálně. „Rychlá“ pumpa je pouze spínána pro oba směry čerpání, ale „pomalá“ pumpa umožňuje navíc i dva režimy činnosti. Prvním je přednastavená rychlost čerpání např. $10 \mu\text{l} \cdot \text{min}^{-1}$ a druhým je maximální rychlost přibližně $100 \mu\text{l} \cdot \text{min}^{-1}$.



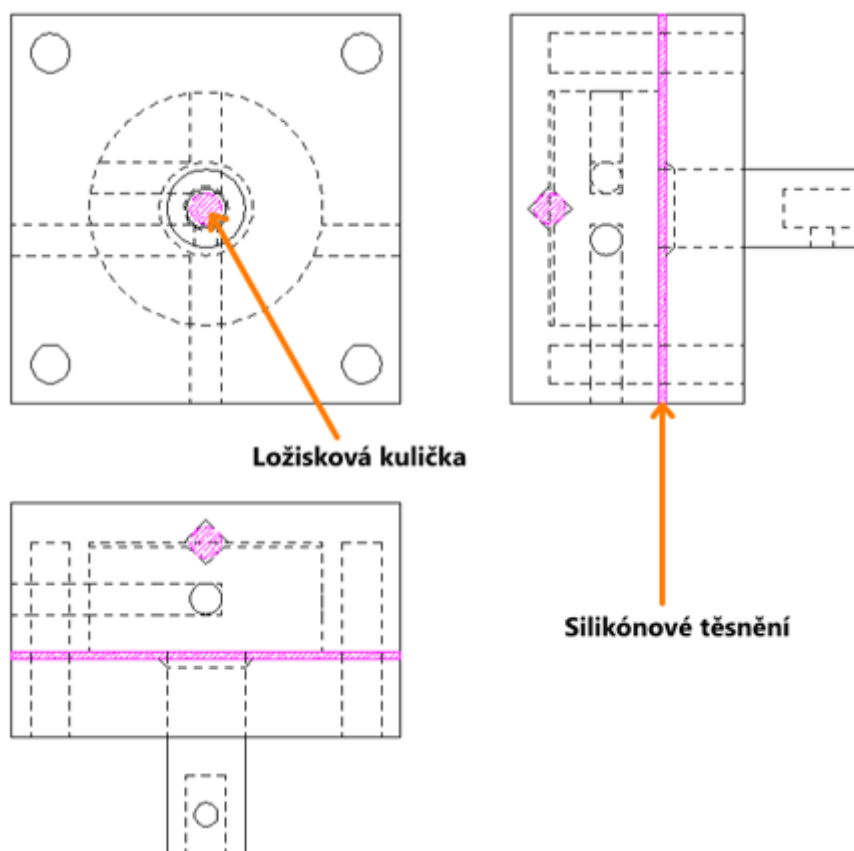
Obr. 24: „Pomalá“ pumpa.



Obr. 25: „Rychlá“ pumpa.

Nutnost obsluhy dvou pump si vyžádala navrhnout i řešení, jak přepínat nebo přiřazovat jednotlivé pumpy k jednomu přečerpávacímu hrotu, tak aby byla zajištěna

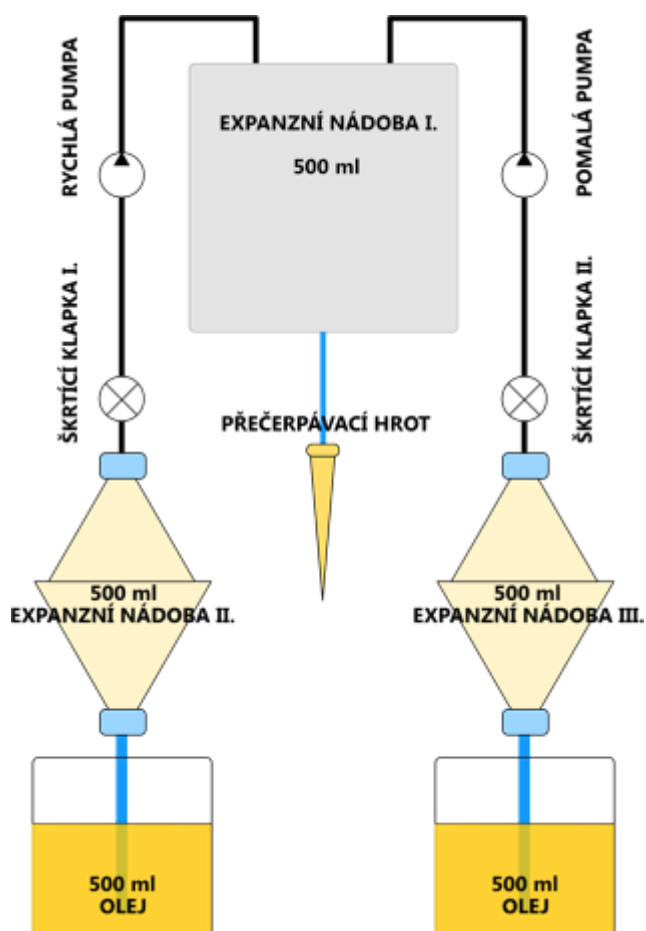
stoprocentní těsnost systému v obou směrech čerpání. Prvním návrhem bylo vyrobení dvouramenného L-ventilu (obr. 26), který bude řízen krokovým motorem a umožní tak přiřazovat vstup pumpy k čerpacímu hrotu.



Obr. 26: Návrh dvojramenného ventilu.

Výroba tohoto ventilu je ovšem dost náročná vzhledem k požadované přesnosti šoupátka a problematická je i těsnost systému, vzhledem ke stárnutí a opotřebení materiálu těsnění.

Druhý alternativní návrh (obr. 27) přinesl řadu výhod a odstranil i potřebu mechanického ventilu. Zakládá se na principu tří expanzních nádob a čerpání provozní kapaliny o vyšší hustotě. Všechny tři expanzní nádoby mají stejný objem, který je navíc větší než maximální čerpané množství míchaného roztoku, a proto nikdy nedojde ke styku pumpy ani provozní kapaliny s čerpaným roztokem. Výstup každé pumpy je napojen na expanzní nádobu, která ústí v olejové lázni a vstup obou pump je sloučen v třetí společné expanzní nádobě. Při čerpání dochází k nasávání jak provozní kapaliny tak roztoku, ale protože roztok má nižší hustotu, dochází k rychlejšímu čerpání roztoku než provozní kapaliny. Důsledkem tohoto uspořádání je stoprocentní těsnost při čerpání, nezávislost na těsnosti jednotlivých pump a možnost upravovat odpor výstupního členu škrticími klapkami. Realizovaný systém tří expanzních nádob je dokumentován na obr. 28.

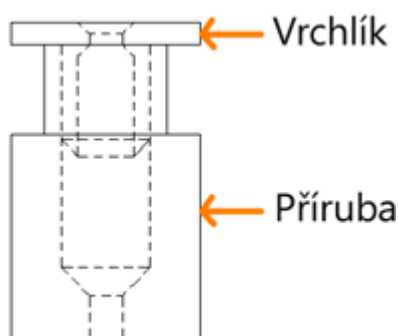


Obr. 27: Teorie tří expanzních nádob.



Obr. 28: Realizace expanzních nádob.

Zajištění čerpání roztoků z ampulí se děje zasunutím přečerpávacího hrotu do ampule, ale čerpání roztoků z objemových nádob (pufr, oplach) a vyčerpávání roztoku z reakční nádoby musí být realizováno přes silikónovou hadici, kvůli nedostačující délce posunu hrotu v ose y a nepřístupnosti nádob. Aby bylo možné čerpat i přes silikónovou hadici byl vyvinut systém přečerpávací spojky (obr. 29). Těsnost spojky je

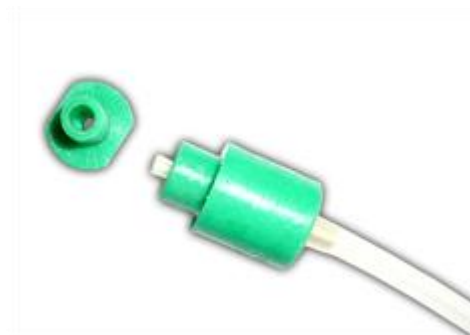


Obr. 29: Návrh přečerpací spojky.

zajištěna dostatečným zasunutím přečerpávacího hrotu do silikónové hadice upevněné v plastové přírubě. Přesnost zasunutí je řízena naváděcím trychtýřem na konci hadice uvnitř příruby a upevnění spojky ve stojanu je realizováno závitovým vrchlíkem spojky. Utažením vrchlíku se dá regulovat i těsnost spojky. Hadice vložená do nádoby a hadice ve spojkce je propojena mosaznou průchodkou. Jednotlivé části spojky jsou dokumentovány na obr. 30 a jedna složená spojka je pak na obr. 31.



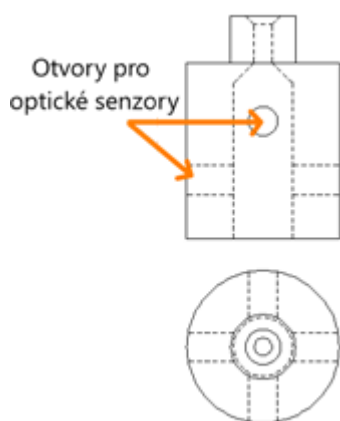
Obr. 30: Součásti přečerpávací spojky.



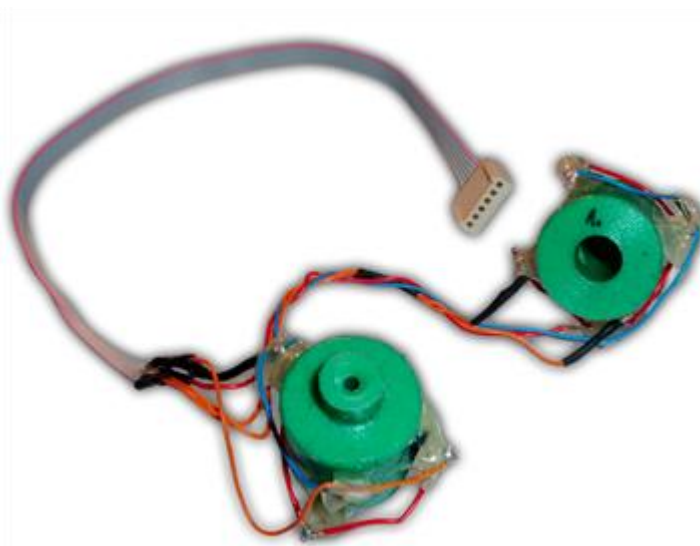
Obr. 31: Složená přečerpávací spojka.

4.4 Zpětná vazba čerpání

Protože jsou primárně čerpány poměrně malé množství roztoků, v řádech μl , je vhodné tyto manipulace určitým způsobem monitorovat. Proto byl navržen systém průchodky s optickou kontrolou odkapávání. Průchodka je navržena tak, aby při plném zasunutí přečerpacího hrotu, byl hrot v optimální pozici pro měření kapek a navíc, aby nedocházelo ke stékání roztoku po stěnách průchodky. Ovládání a snímání signálu z páru optických senzorů umístěných v průchodce se děje jak hardwarově, tak softwarově. Na obr. 32 je zobrazen návrh průchodky a na obr. 33 je fotografie realizovaného páru průchodek pro sledování manipulace s roztoky v reakční nádobě a odpadišti.



Obr. 32: Návrh optické průchodky.



Obr. 33: Realizace páru průchodek.

5 Výběr hardwarového vybavení

Výběr hardwarového vybavení sebou nese důkladnou studii dostupných produktů, které splňují požadavky pro splnění zadané problematiky. Při výběru hraje důležitou roli jak dostupnost zvolených součástí, tak i jejich poměr cena/výkon. Protože navrhované zařízení se skládá z většího počtu dílčích celků, bylo nutné tyto úvahy zohlednit u každé části individuálně.

První oblastí zájmu se stal výběr mikrokontroléru a součástek pro realizaci komunikace. Protože v bakalářské práci autora se osvědčila práce s mikrokontroléry firmy ATMEL, byla volba řídicího prvku do určité míry ulehčena. Provedení komunikace vyžadovalo oproti bakalářské práci zásadní změnu ve směru modernizace. Nabízely se dvě možnosti a to komunikace přes USB nebo vývoj IP zařízení. Ve volbě komunikace hrála velkou roli uživatelská přívětivost a efektivita při pohledu na zařízení jako celek.

Další oblastí zájmu byla jednotka potenciostatu, která musí zajistit správnou komunikaci s čipem potenciostatu, který byl vyroben v rámci projektu IMAM (GA ČR 102/03/0619 – IMAM Inteligentní mikrosenzory a mikrosystémy pro měření, regulaci a životní prostředí). Zde se volba součástek soustředila na vhodné AD a DA převodníky a referenční zdroj napětí.

Řídicí jednotka, která ovládá všechny mechanické a kontrolní prvky je složena v zásadě z běžných součástek, které vycházejí i ze zkušeností získaných v bakalářské práci, a proto se řešila hlavně optimalizace pro konkrétní typ obvodových bloků.

Při volbě součástek pro ovládací panel hrála důležitou roli schopnost uživatele dostatečně informovat o aktuálním prováděném stavu pomocí výstupu na LCD displeji a kontrolními led diodami s ohledem na estetickou kompatibilitu s navrženým designem celého zařízení. Na čelním panelu byly instalovány i konektory pro připojení senzorů. Při volbě konektorů bylo vhodné zvolit mechanicky robustní konektory, které zajistí v praxi jednoduchou práci se senzory.

5.1 Výběr mikrokontroléru

V bakalářské práci autora byl použit mikrokontrolér ATmega32, a protože práce s touto rodinou mikrokontrolérů firmy ATMEL se osvědčila, byl výběr vhodného mikrokontroléru soustředěn opět tímto směrem. Hlavním, kritériem pro výběr bylo dostatečné množství bran, velikost FLASH paměti a cena. Jako ideální volbou pro tuto práci se jevil mikrokontrolér ATmega128 [19], který obsahuje téměř 7 bran, 2x USART a 128 KB FLASH paměti. Na první pohled předimenzovaný mikrokontrolér se využil téměř bezzbytku, ale postupně byly některé brány uvolněny optimalizací a jsou proto přístupné pro potřeby možného dalšího vývoje.

5.1.1 Základní informace o ATMega

Firma Atmel s rodinou AVR [20], ukazuje, že RISC architektura nemusí být používána jen u výkonových procesorů pro pracovní stanice nebo 32-bitových mikroprocesorů pro intenzivní početní úkony, ale má smysl i u 8-bitových mikroprocesorů. AVR nabízí nejen jednocyklové instrukce, vyšší taktovací frekvence spojené s vyšším pracovním výkonem, ale i efektivní optimalizace překladu. Rodina AVR je konkurentem na trhu některých dobře zavedených mikroprocesorových rodin, jako jsou 8-bitové rodiny 6805 a 68HC11 od Motoroly stejně jako 8051 od Intelu, jejichž čipy vyrábějí různí vlastníci licencí včetně Atmelu. Hlavním konkurentem je firma Microchip s moderní 8-bitovou rodinou PIC.

5.1.2 Architektura AVR mikrokotrolérů

Jádro AVR [20] se podobá jádru většiny RISC-procesorů dostupných na trhu. Skládá se ze 32 stejných 8-bitových registrů, které mohou obsahovat jak data, tak adresy. Posledních 6 registrů se může použít ve dvojici jako ukazatele adresy pro nepřímé adresování paměti dat. Tyto registry označované písmeny X, Y a Z dovolují libovolné ukládací operace (LOAD/STORE). Programátor má například na výběr, zda ukazatel adresy bude po zpracování určité instrukce inkrementovat nebo před zpracováním této instrukce dekrementovat. Užitečné je pro adresování využít možnosti 6-bitového posunu v ukazateli adresy v dvojitých registrech Y a Z.

AVR architektura má 5 adresovacích módů pro paměť dat:

- přímé adresování,
- nepřímé adresování s posunutím (6-bitový posun),
- nepřímé adresování,
- nepřímé adresování s dekrementací ukazatele adresy před zpracováním instrukce,
- nepřímé adresování s inkrementací ukazatele adresy po zpracování instrukce.

Registry jsou zobrazovány přímo v adresovém prostoru dat. Prvních 32-bytů paměti (0x00 až 0x1F) odpovídá registrům R0 až R31, ty jsou také přímo zamapovány do paměti dat. Proto je možno s každým registrem zacházet použitím standardních odkazů bez toho, aby programátor potřeboval znát řídicí instrukce registrů. AVR používá Harvardskou architekturu - paměť programu a paměť dat jsou odděleny. Paměť programu je přístupná pomocí dvoustupňové pipeline. Když se určitá instrukce začíná vykonávat, další instrukce je připravována k zpracování. Tato konstrukce dovoluje zpracovávání instrukcí vždy v jednom hodinovém cyklu.

Zpracování operandů probíhá tak, že během jednoho taktovacího cyklu se přivedou oba zdrojové operandy z pracovních registrů, uskuteční se potřebná operace a výsledek se uloží zpátky do registrů. Jedná se výhradně o operace typu register - register, na kterých je architektura AVR založena a důsledně je dodržuje podle modelu LOAD/STORE.

Většina všech instrukcí je 16-bitových. Pouze 4 instrukce jsou 32-bitové, které jsou však omezeny dovoleným 16-bitovým adresováním. Celá instrukční sada s maximálně 118 instrukcemi nutí také ke kompromisům. Většina instrukcí v instrukční sadě má přímý a jednocyklový přístup do všech registrů. Výjimku tvoří pouze aritmetické a logické instrukce jako SBCI, SUBI, CPI, ANDI a ORI mezi konstantou a registrem a instrukce LDI. Jedná se tedy hlavně o zpracování přímých datových typů. Jen výše uvedené instrukce akceptují přímé hodnoty a mohou být používány pouze v horní polovině registrů (R16 až R31). Ještě je možné pro zpracování přímých datových typů použít instrukce ADIW a SBIW, které jsou přístupné pouze v posledních 8 registrech (R24 až R31).

Pro podmíněné větvení a skoky lze použít různé instrukce. Na každý z osmi příznaků ve Status registru připadají nejméně 2 instrukce podmíněného skoku (např. BREQ, BRCS atd.). Další instrukce pro skoky je RJMP (12-bitová), kterou lze posunout relativně kód o 2 Kbit a také absolutní skoková instrukce IJMP, která provede skok na danou adresu. Zajímavé jsou také mnohé "Skip" instrukce (SBRC, SBRS, SBIC a SBIS), které umožňují podmíněný skok přes následující instrukci.

Pro aritmetické a logické operace obsahuje instrukční soubor klasické instrukce sčítání ADD, ADC, odečítání SUB, SUBI, SBC, SBCI, logické instrukce jako AND, OR, EOR atd. Instrukce MUL má podle definice vynásobit spolu dva libovolné 8-bitové registry během dvou taktovacích cyklů a 16-bitový výsledek se má uložit do registrů R0 a R1.

AVR-architektura disponuje také instrukcemi pro bitové operace a se 16 definovanými instrukcemi je možno nastavit nebo vymazat každý příznak (Flag) ve Status registru. Stejný efekt vznikne také pomocí logických operací, kde se tímto způsobem dosáhne v mnoha aplikacích značného zjednodušení. Například instrukce SER a CLR nastaví eventuálně smažou najednou celý registr a instrukce SBR a CBR mají vliv na vybrané bity v jednom registru.

6 Volba komunikačního rozhraní s přípravkem

Hlavní důvod, proč bylo nutné modernizovat řešení komunikace vycházející z bakalářské práce autora, je to, že rozhraní RS232 již přestává být dostupné na většině nových PC. Původní zařízení v bakalářské práci navíc využívalo poměrně složitý systém pro programování mikrokontroléru s využitím ISP programátoru. Protože tento programátor komunikuje rovněž přes rozhraní RS232, bylo nutné komunikační linky obvodově oddělit.

Rozhraní USB sebou přináší nejen odstranění problému s absencí portů RS232 a oddělením komunikačních linek, ale i nutnost vytvářet stabilizovaný zdroj. Celá aplikace může být napájena přes USB a v případě, že tento zdroj bude nedostatečný, stačí pouze připojit externí stabilizovaný zdroj napětí.

Z pohledu uživatele se rozhraní USB jeví jako přívětivější hlavně s ohledem na to, že takto navržené zařízení se může chovat způsobem Plug and Play (tedy zapoj a pracuj/hraj si). Softwarově se dá zajistit, aby se uživatel nemusel starat o žádné nastavení spojené s komunikací. Moderní operační systémy jako např. WINDOWS 7 již nevyžadují ani manuální instalaci ovladače pro virtuální COM port. Navíc přes toto rozhraní lze velmi snadně softwarově ladit embedded software zařízení.

V případě vývoje IP zařízení se stojí před problémem složitosti prvotního nastavení spojení a problematiky vnitřních sítí a jejich ochrannými mechanismy. Zařízení, ke kterému lze přistupovat přes Internet, což je v tomto případě prioritní, musí mít veřejnou IP adresu a tím se může stát potenciálně slabým místem v ochraně přístupu do vnitřních sítí. Při potřebě ovládání přes Internet se obvykle nelze vyhnout ani instalaci podpůrného softwaru na straně uživatele vzdáleného počítače, což zcela popírá základní myšlenku vyvíjeného komunikačního protokolu. Navíc vyvíjené zařízení se neobejde bez obslužného počítače kvůli nutnosti spuštění kalibračního softwaru. V případě realizace systému s komunikací přes IP se nelze obejít ani bez rozbočovacího prvku jako je HUB či SWITCH, protože je nutné využívat webkameru a připojení do sítě pro obslužný počítač. Vzhledem ke značné problematice tohoto řešení bylo zvoleno rozhraní USB.

6.1 Programování mikrokontroléru ATmega přes USB

Jak již bylo zmíněno, zvolený způsob komunikace umožňuje i snadné ladění softwaru mikrokontroléru. Pro programování mikrokontroléru ATmega přes USB je potřebné:

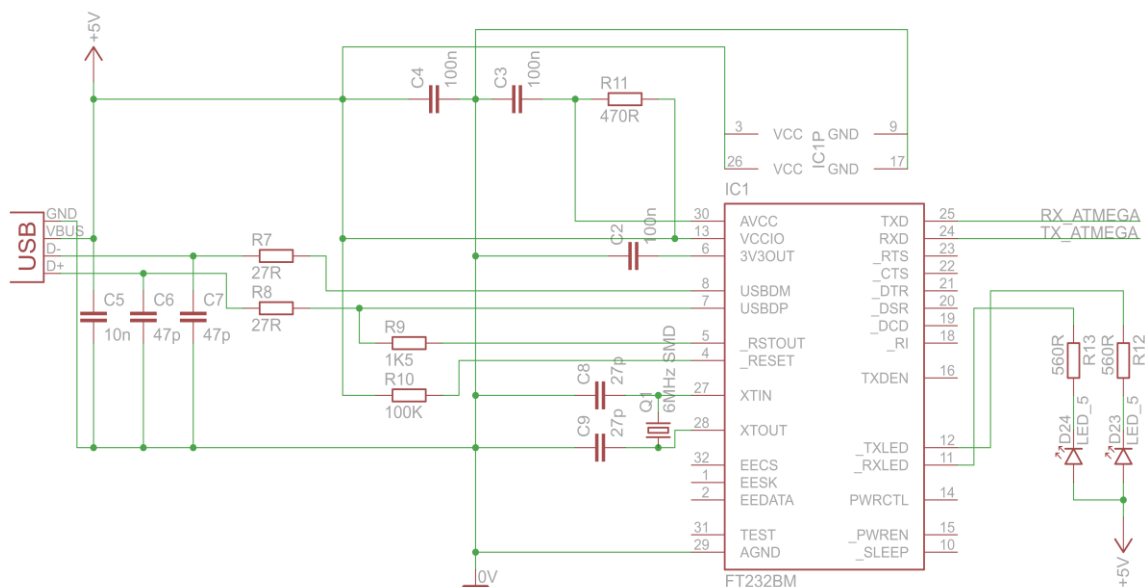
- převodník USB-RS232,
- ISP programátor,

- bootloader (zavaděč),
- AVR studio.

Programování je založeno na využití bootloaderu, který umí zavádět řídicí program přes UART. Převod RS232 na USB je zajištěn integrovanou součástkou FT232BM [21] od firmy FTDI a vytvořením virtuálního COM portu na obslužném počítači. K vývoji a nahrání řídicího programu je s výhodou použito AVR studio [22] od firmy Atmel.

6.2 RS232-USB pomocí FT232BM

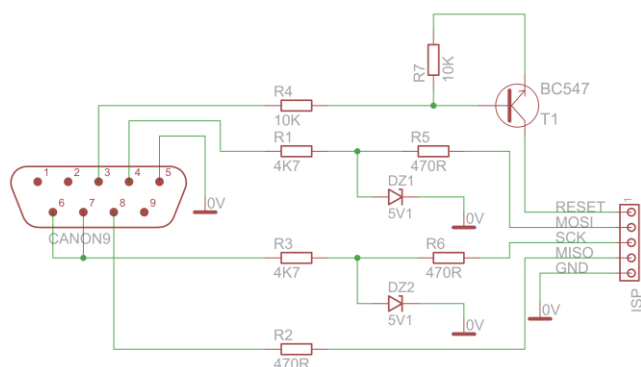
FT232BM je integrovaná SMD součástka, která dokáže převést USB na komunikaci přes UART s využitím virtuálního COM portu. Virtuální COM port je volně šiřitelný software firmy FTDI a není proto nutné zakoupit licenci na provoz zařízení využívající rozhraní USB. Zapojení FT232BM je jednoduché a je uvedeno na obr. 34.



Obr. 34: Zapojení FT232BM.

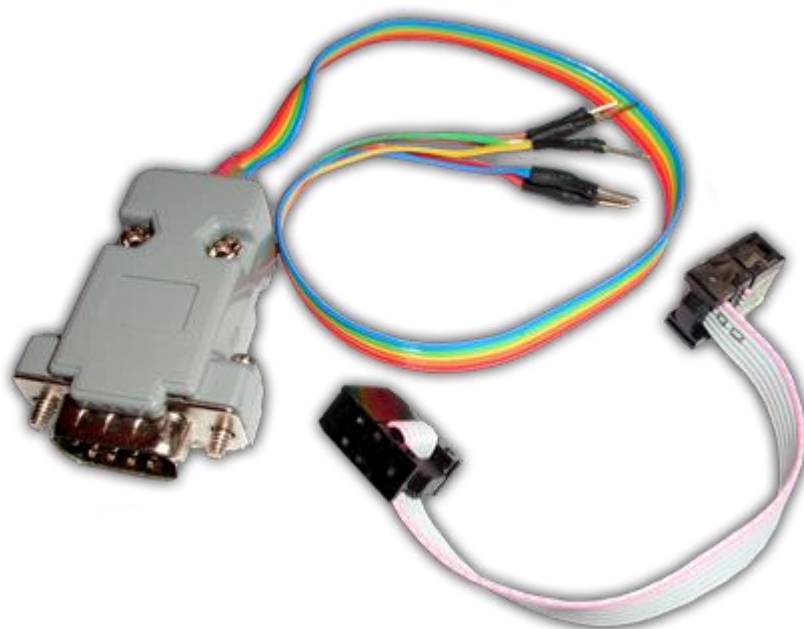
6.3 ISP programátor

Před samotným zahájením programování s bootloaderem je nutné tento program vypálit do mikrokontroléru, a proto je zde použit externí ISP programátor a PonyProg [23]. Zapojení ISP programátoru je uvedeno na obr. 35.



Obr. 35: Zapojení ISP programátoru.

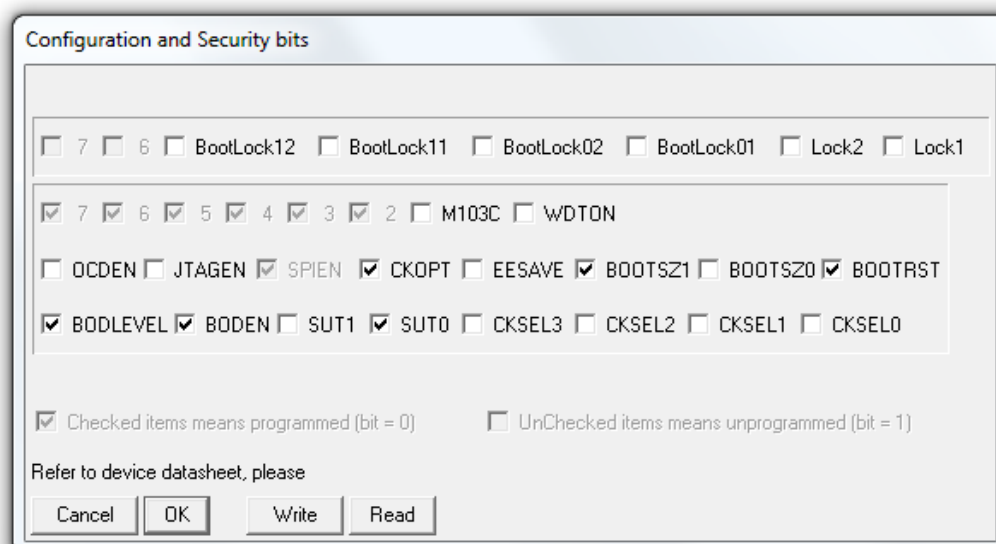
Protože ISP programátor se v ideálním případě využije jen jednou, tedy jen pro zavedení bootloaderu, není nutné, aby byl pevnou součástí obvodového řešení, a provede se pouze jako externí zařízení (obr. 36), které se připojí do konektoru, na níž budou vyvedeny potřebné piny mikrokontroléru.



Obr. 36: Externí ISP programátor s redukcí.

Zapojení kabelů: **modrý** – GND, **žlutý** – RESET, **červený** – SCK, **zelený** – MISO, **oranžový** – MOSI.

Pro správnou funkci mikrokontroléru s bootloaderem je potřebné navíc nastavit fuse bity dle daného typu ATmega. Pro ATmega128 [19] může tato konfigurace vypadat jako na obr. 37. Jednotlivé typy mikrokontrolérů ATMEL se mohou v nastavení fuse bitů lišit, ale obvykle je nutné nastavit bity BOOTSZ1, BOOTSZ0 a BOTRST. Těmito bity se alokuje potřebný prostor v paměti a sdělí se mikrokontroléru, že se má po resetu spustit zavaděč.



Obr. 37: Fuse bity pro ATmega128 s bootloaderem.

6.4 Bootloader

Bootloader je program o velikosti 512 bitů – 1 Kbit, který se vypaluje do speciální části paměti mikrokontroléru a s jeho pomocí je umožněno měnit řídicí program. Program je založen na komunikaci přes UART a může se lišit ve své inicializaci a způsobu vypalování řídicího programu. Zavaděč může pro vypalování využívat na straně PC buď vlastní softwarovou podporu, nebo může komunikovat přímo s vývojovým prostředím AVR studio [22]. Bootloader sám o sobě zahrnuje nastavení pro konkrétní mikrokontrolér. Uvádí se jak jeho typ a popis, tak i použitá frekvence a přenosová rychlost COM portu. Protože navržená aplikace pracuje na 16 MHz je přenosová rychlost nastavena na 38400 a je tak zajištěna minimální chybovost při komunikaci.

6.4.1 Bootloader s hardwarovou inicializací

První typ zavaděče s podporou AVR studia využívá uživatelem řízené spuštění rutiny pro vypalování řídicího programu. Na jeden z portů je vyvedena signalizační dioda a druhý port sleduje zvolenou úroveň signálu. Dle zvolené úrovně se rozhodne o tom, zda-li se spustí rutina čekající na inicializaci ze strany PC nebo jestli se přejde rovnou k vykonávání řídicího programu. Pokud je vybrána první možnost začne signalizační dioda blikat po dobu 10 s a pokud se neprovede inicializace, přestoupí se k vykonávání řídicího programu. Na obr. 38 je znázorněno jedno z možných zapojení pro bootloader s hardwarovou inicializací.

7 Realizace obvodových bloků

Zařízení se skládá z několika celistvých obvodových bloků, které řeší konkrétní části zařízení. Blokový návrh se zvolil hlavně pro možnost snadné úpravy nebo opravy konkrétního bloku. Byly navrženy a realizovány čtyři obvodové bloky:

- sběrnice pro mikrokontrolér ATmega128 a redukce pro osazení mikrokontroléru,
- řídicí jednotka,
- jednotka potenciostatu,
- ovládací panel.

Všechny obvodové bloky byly realizovány formou dvouvrstvé DPS s kombinovanou montáží, přičemž důraz se kladl na maximální optimalizaci pro povrchovou montáž.

7.1 Sběrnice pro mikrokontrolér ATmega128

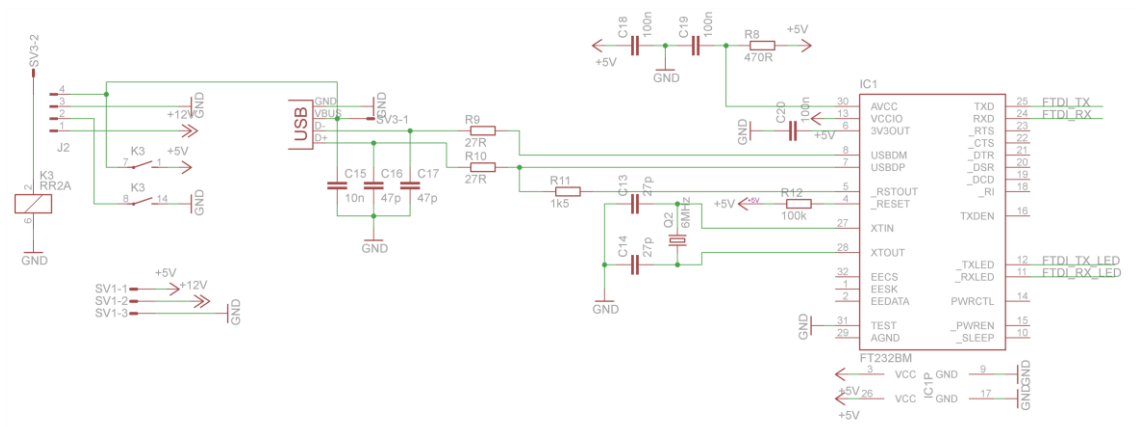
Jak již název napovídá, tento blok slouží pro rozvod signálových cest řídicího prvku ATmega128. Snadnost realizace sběrnice byla docílena vytvořením redukce pro mikrokontrolér (obr. 42), která se do sběrnice (obr. 43) zasouvá prostřednictvím precisní patice. Redukce navíc obsahuje veškeré součástky nutné pro funkci mikrokontroléru (resetovací obvod, krystal, reference pro AD převodníky, napájecí větve), ale taky patici pro připojení externího ISP programátoru. Obvodový blok kromě rozvodů signálových cest zajišťuje i rozvod napájení +5 V a +12 V a blok pro komunikaci přes USB (obr. 40). Na DPS sběrnice je umístěn i blok pro zapínání přístroje kolébkovým spínačem. Kolébkový spínač ovládá relé, které umožňuje sepnutí +5 V větve z USB do všech obvodových bloků a zapnutí externího zdroje napětí.



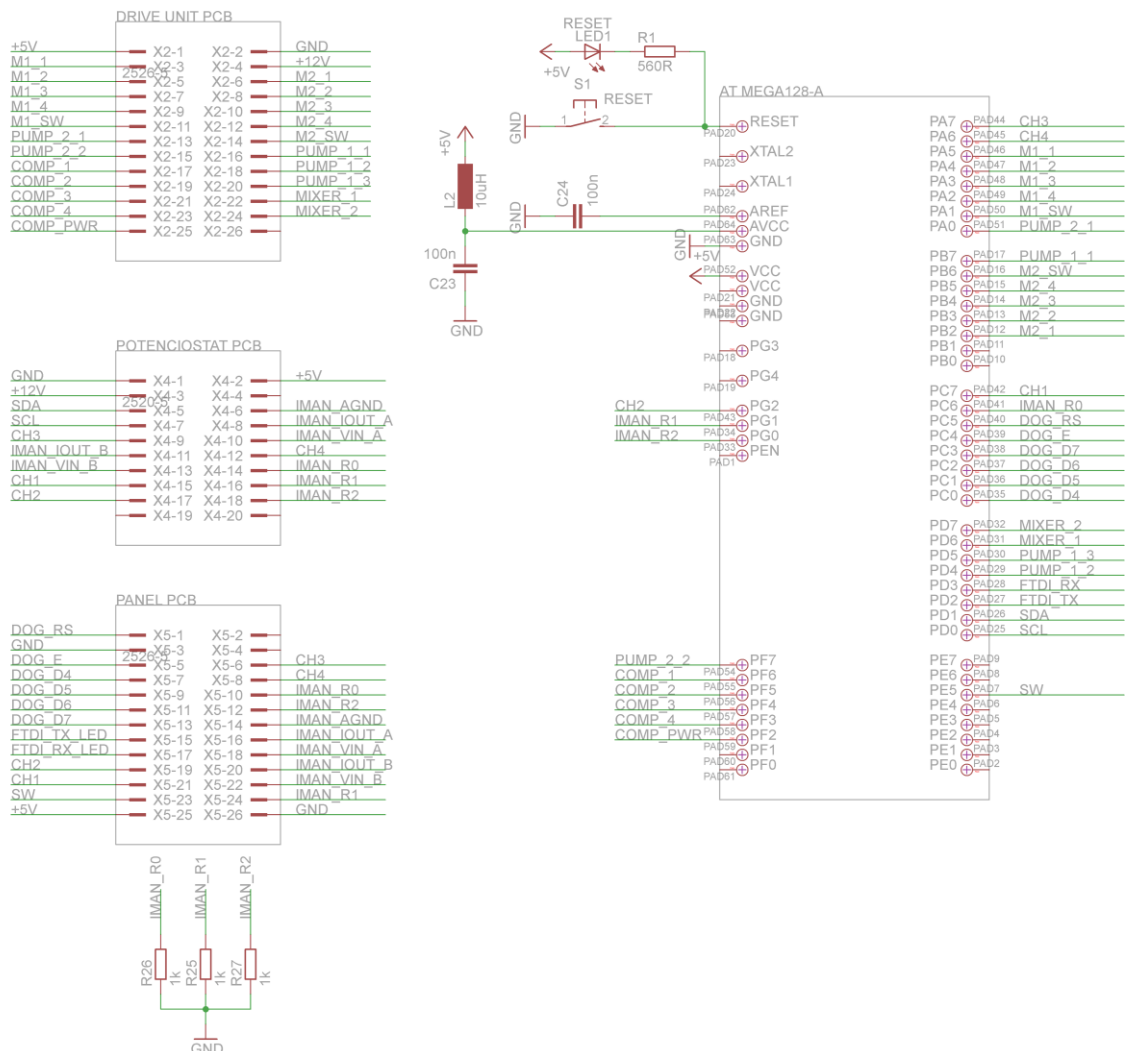
Obr. 39: Externí zdroj pro podporu +5 V větve a zdroj +12 V větve.

Externí zdroj (obr. 39) proudově posiluje +5 V větev z USB a umožňuje i odběr +12 V větve. Na sběrnici (obr. 41) jsou konektory pro připojení DPS ovládacího panelu, řídicí jednotky a jednotky potenciostatu. Pro případ potřeby je na DPS vyveden i konektor pro odběr napájecích napětí a analogové země. Redukce pro mikrokontrolér je realizována tak, aby byly v případě potřeby snadně přístupné všechny piny mikrokontroléru a to jak z důvodu měření

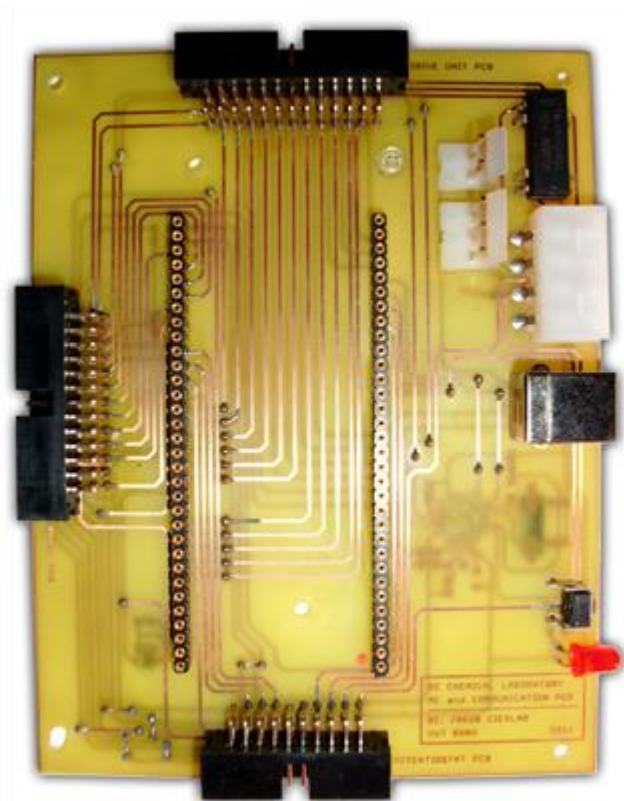
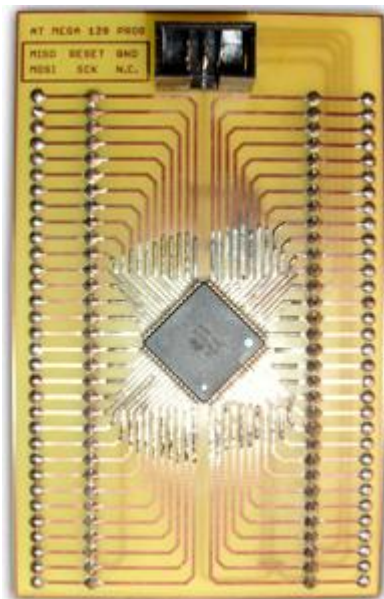
signálových cest, tak i pro možnost přístupu k nevyužitým portům.



Obr. 40: Zapojení čipu FTDI a rozvod napájecích napětí.

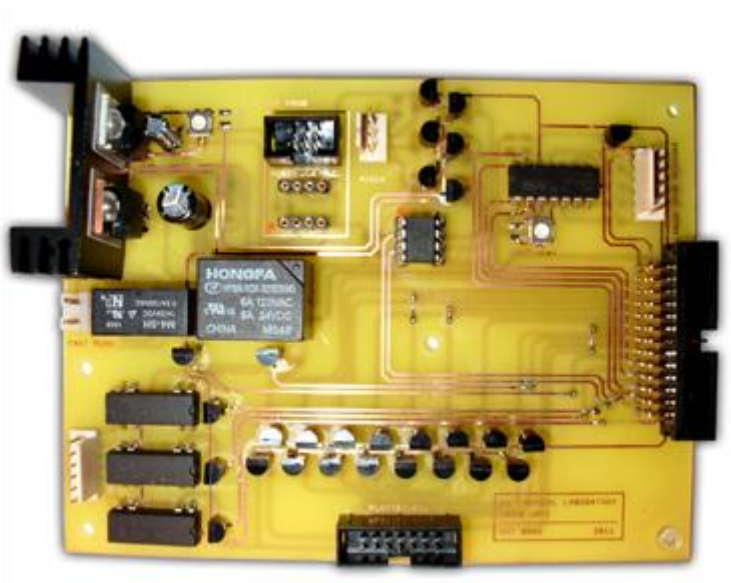


Obr. 41: Zapojení portů mikrokontroléru ATmega128.



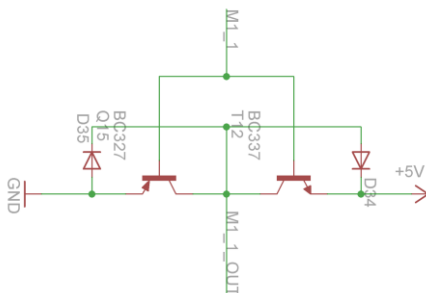
7.2 Řídicí jednotka

Řídící jednotka (obr. 44) obsahuje obvodové obloky pro ovládání všech mechanických součástí, ovládání a vyhodnocování signálu z optických senzorů pro zpětnou vazbu čerpání roztoků, dále rozvádí napájecí větve a signálové cesty a obsahuje mikrokontrolér ATTiny13 pro řízení míchadla roztoků.



7.2.1 Ovládání krokových motorů

Pro ovládání jednotlivých cívek krokových motorů instalovaných na plastovém běžci byly zvoleny tranzistorové H-můstky na obr. 45. Propojení DPS umístěné na plastovém běžci a řídicí jednotky je realizované plochým kabelem, jímž jsou vedeny i signály od kalibračních spínačů a napájecí a zemní větve.



Obr. 45: H-můstek pro ovládání cívky krokového motoru.

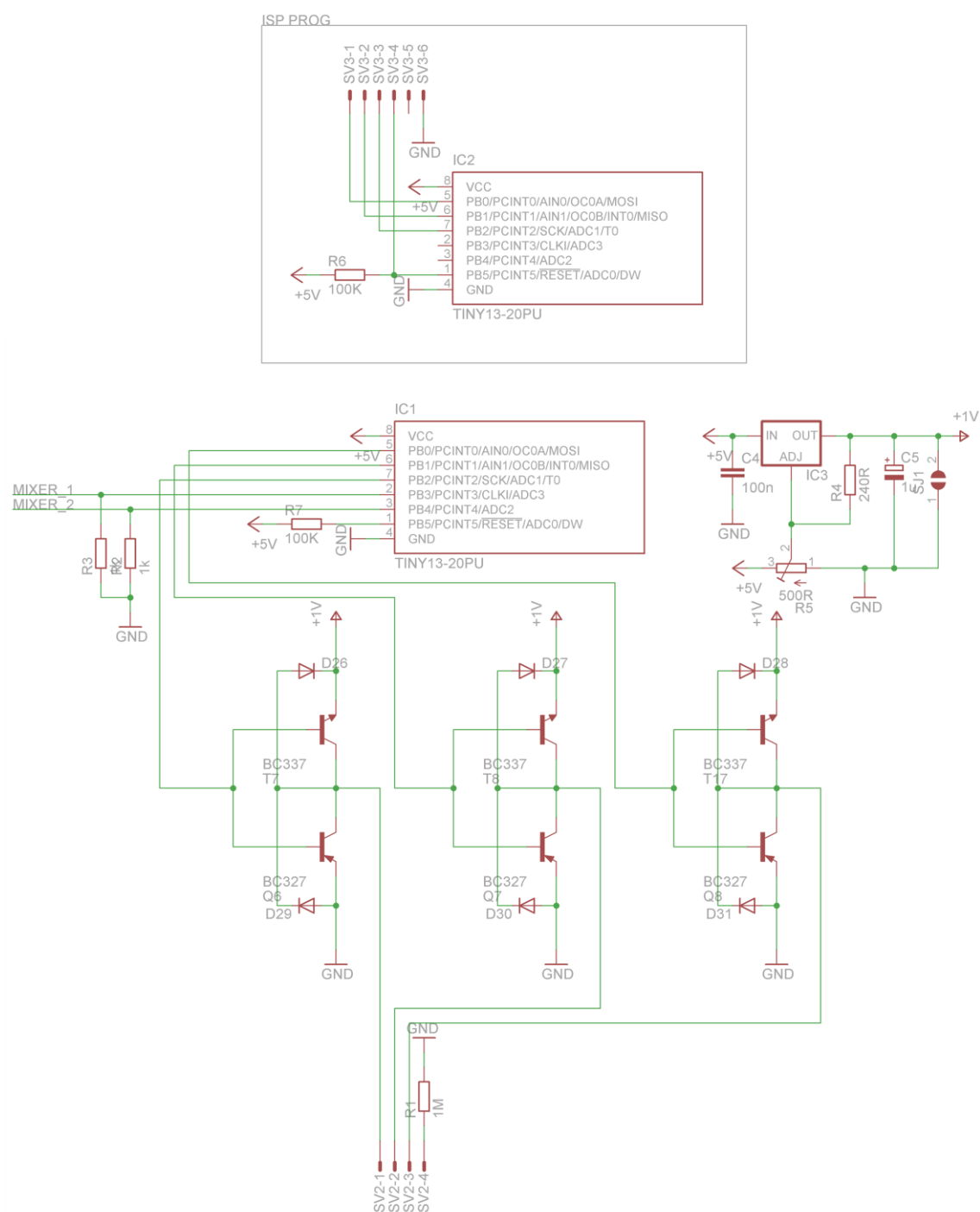
7.2.2 Ovládání míchadla roztoků

Míchadlo roztoků (obr. 46) bylo realizováno formou elektromagnetické střelky, jejíž pohyb v reakční nádobě je řízen statorem z motoru 3,5'' disketové mechaniky. Míchání roztoku by mělo být prováděno v průběhu jeho přípravy a případně i v průběhu proměření senzoru. Protože současné ovládání míchadla a rychlý sběr dat při proměření senzoru by byl těžce realizovatelný jedním mikrokontrolérem, byl do bloku pro ovládání míchadla (obr. 47) zařazen jednoduchý mikrokontrolér ATTiny13-20PU [24]. Řízení by šlo realizovat i jednodušším způsobem např. pomocí čítače, ale vzhledem k velmi nízké ceně zvoleného mikrokontroléru a možnosti volby i více rychlostí míchání byla zvolena tato varianta. Protože zvolený mikrokontrolér má pouze 6 pinů, na nichž jsou sloučeny různé funkce, nelze jej při plném využití všech pinů programovat přímo v obvodu a to i přes podporu ISP programátoru. Z tohoto důvodu byl na DPS řídicí jednotky vytvořen jednoduchý obvod pro programování mikrokontroléru. Ovládání míchadla je založeno na zapojení z tranzistorových H-můstků. Kvůli malému vnitřnímu odporu cívek muselo být zapojení doplněno o regulovaný zdroj napětí pro vytvoření $U = +1$ V napájecí větve. Rychlost míchání je ovládána dvěma piny mikrokontroléru. Kombinace vstupních signálů umožňuje nastavit čtyři režimy činnosti:

- vypnutí míchadla,
- pomalé míchání,
- střední míchání,
- rychlé míchání.



Obr. 46: Míchadlo roztoků.

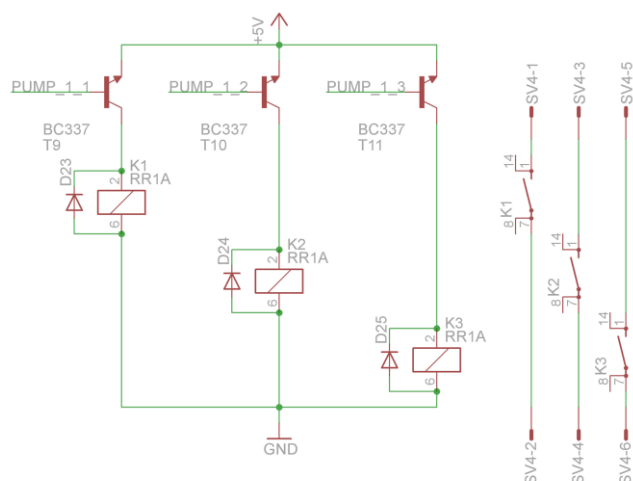


Obr. 47: Zapojení bloku míchadla.

7.2.3 Ovládání pump

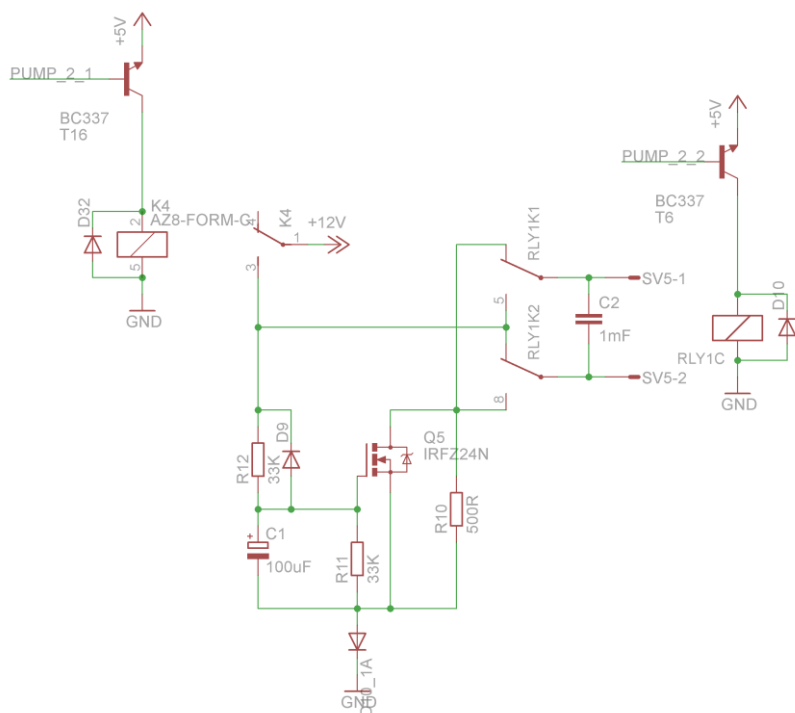
Obě zvolené pumpy pro čerpání roztoků jsou původně ovládány manuálně. Pro zajištění digitálního řízení musel být vyvinut blok, který umožní jejich řízení mikrokontrolérem.

„Pomalá“ pumpa byla rozebrána a studií vnitřních obvodů bylo zjištěno, kterými signály se pumpa ovládá. Signálové vodiče byly vyvedeny a paralelním zapojením tří relé se umožnilo digitální řízení (obr. 48).



Obr. 48: Blok pro ovládání „pomalé“ pumpy.

Polarita napájecího napětí „rychlé“ pumpy je ovládána třípolohovým spínačem, proto se pro ovládání polarity použilo dvakrát přepínací relé a pro zapínání a vypínání motoru jedenkrát spínací relé. Motor „rychlé“ pumpy je navinut pro napájecí napětí +12 V. Pro napájení tohoto motoru byla zvolena +12 V větev externího zdroje, ale vzhledem k velké indukčnosti cívky rotoru docházelo při zapínání a hlavně při vypínání, ke vzniku proudových špiček, které se přenášely společnou zemí i na +5 V větev. Proudové špičky způsobovaly resetování mikrokontroléru ATmega128, a proto byl do bloku pro ovládání pumpy zařazen i soft-start (obr. 49), jímž se řídí postupný rozběh a zastavení motoru. Soft-start redukuje proudové špičky a resetování mikrokontroléru se tímto způsobem odstranilo.



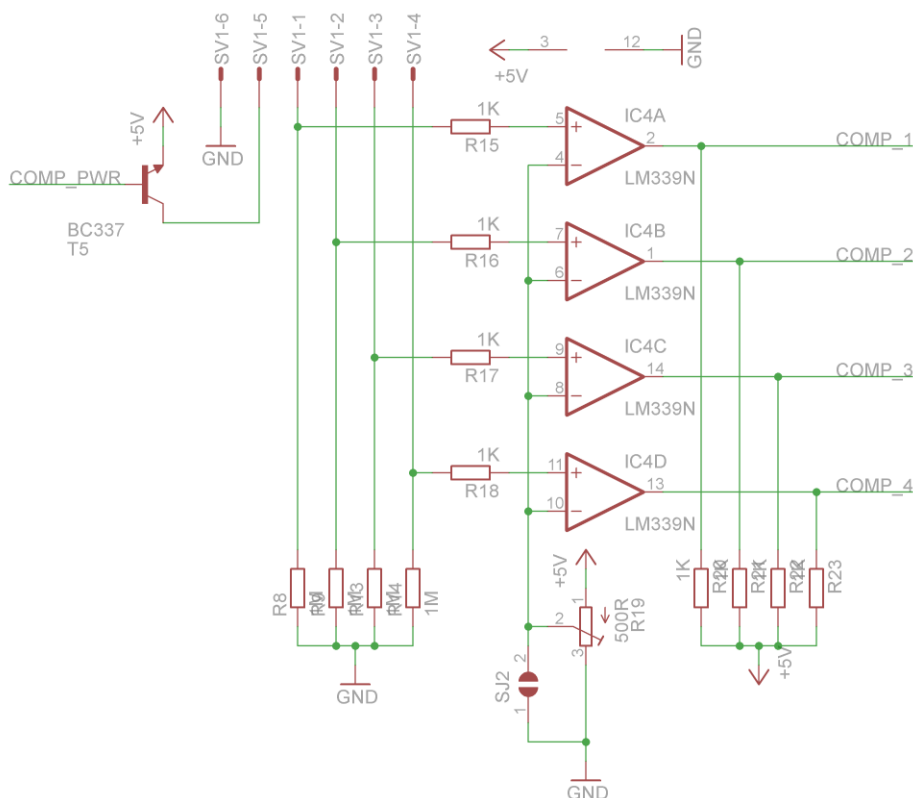
Obr. 49: Blok pro ovládání „rychlé“ pumpy.

7.2.4 Blok komunikace se senzory pro kontrolu čerpání

Při čerpání malých objemových množství roztoků je vhodné určitým způsobem sledovat funkčnost pumpy respektive sledovat odkapávání kapek roztoku z přečerpávacího hrotu. Protože způsobů realizace těchto zpětnovazebních členů je velké množství, byl navržen blok (obr. 50), který dokáže komunikovat se všemi senzory na stejné úrovni.

Blok se skládá ze čtyř komparátorů, které mají společné referenční napětí. Referenční napětí je nastavitelné a výstupy z komparátorů jsou vyhodnocovány mikrokontrolérem ATMegal28. Pro svou jednoduchost byl navržen optický zpětnovazební člen, ale protože některé roztoky mohou být světlem ovlivněny, byl komunikační blok obohacen i o možnost digitálního ovládání zapnutí a vypnutí senzorů.

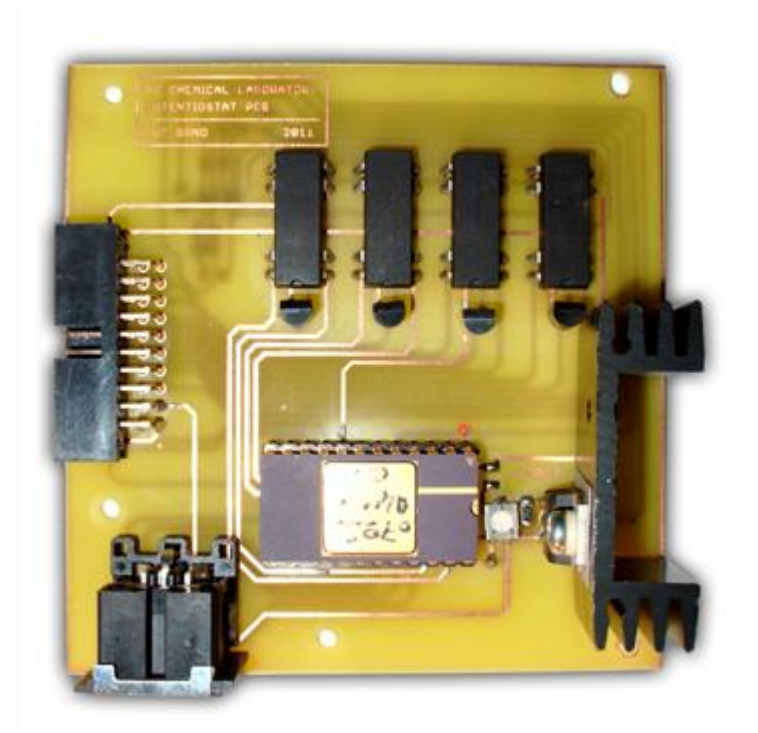
Návrh jiných typů zpětnovazebních členů bude spočívat v tom, aby jejich výstupní signály byly správně vyhodnoceny komparátory tohoto bloku.



Obr. 50: Blok komunikace se zpětnovazebními členy.

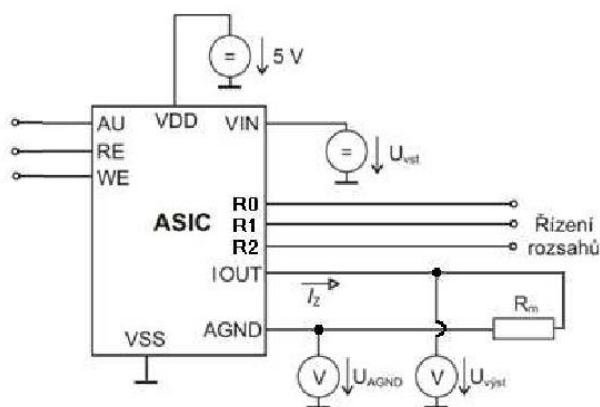
7.3 Jednotka potenciostatu

Jednou z nejdůležitějších součástí celého zařízení je právě jednotka potenciostatu (obr. 51), která zajišťuje proměření testovaných senzorů. Návrh obvodového bloku byl zjednodušen využitím integrovaného čipu potenciostatu IMAM, který byl realizován Ústavem mikroelektroniky (UMEL).



Obr. 51: DPS jednotky potenciostatu.

Potenciostat (obr. 52) je zařízení, které přivádí definované hodnoty napětí na měřený senzor a sleduje jeho proudovou odezvu. Proudovou odezvu pak převádí na odpovídající hodnotu napětí, která je měřena dalším obvodem. IMAM má možnost nastavení rozsahu měřeného proudu a piny pro zapojení referenční (RE), pracovní (WE) a pomocné (AU) elektrody. Čip je navržen tak, aby mohl být integrován přímo na TLV senzor, ale je proveden i ve 24 pinovém pouzdru DIL.



Obr. 52: Zapojení integrovaného obvodu IMAM [25].

Požadavek na realizaci této jednotky (obr. 53) byla schopnost ovládat oba typy provedení čipu. Protože čip IMAM obsahuje dvě jádra potenciostatu je možné navrženým zařízením volit mezi čtyřmi nezávislými kanály. Zapojení je, ale realizováno tak, že lze měřit vždy jen na jednom kanále a výběr kanálu je řešen pomocí čtyř dvakrát přepínacích relé. Čip je řízen 12-bit DA převodníkem a porty mikrokontroléru

7.4 Čelní panel

53



Obr. 54: DPS čelního panelu.

Při návrhu se dbalo na minimalistické a co nejpřehlednější řešení (obr. 55). Na ovládacím panelu je proto umístěn jen třířádkový displej EA-DOGM163, tlačítko pro okamžité zastavení jakékoliv činnosti, kontrolky pro signalizaci napájecího napětí, komunikačních signálů RX a TX, volby vybraného měřicího kanálu a kolébkový vypínač pro zapnutí a vypnutí celého zařízení.



Obr. 55: Detail zařízení – čelní panel.

Do čelního panelu byly instalovány konektory pro připojení propojovacích kabelů (obr. 56) mezi senzorem a jednotkou potenciostatu.



Obr. 56: Konektory pro připojení senzorů.

8 Realizace softwarového vybavení

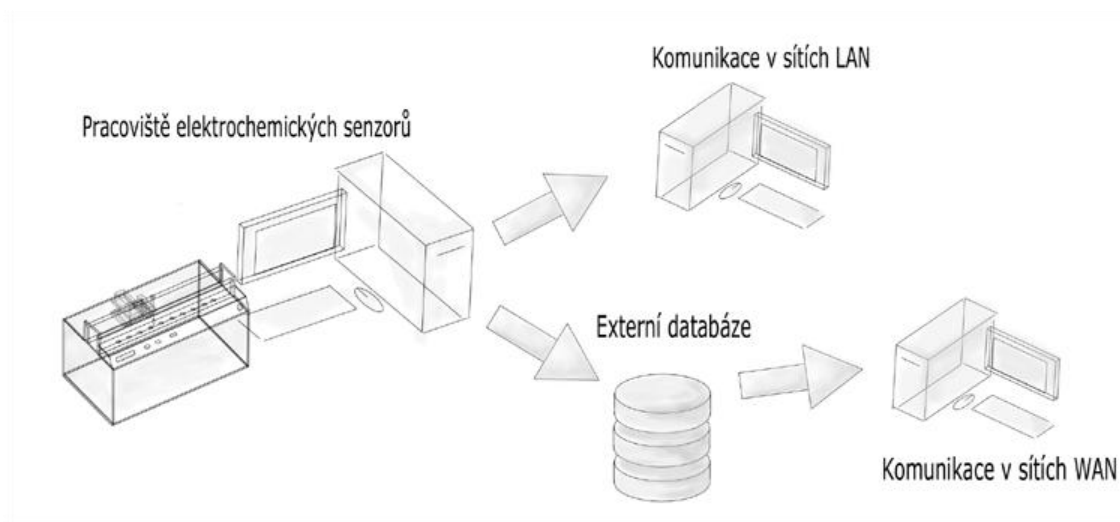
Realizace softwarového vybavení vychází z globálního pohledu na základní požadavky funkčnosti navrhovaného zařízení. Software se proto dělí na tři základní bloky:

- embedded software,
- ovladač pro obslužný počítač,
- uživatelské prostředí internetové aplikace.

Naprogramování jednotlivých bloků se liší jak ve využití vhodných programovacích jazyků, tak i ve struktuře samotného chování a obsahu jednotlivých funkcí. Propojení jednotlivých bloků zajišťuje využití standardizovaných komunikačních protokolů pro použitá komunikační rozhraní. Použitá komunikační rozhraní závisí jak na hardwarové podpoře, tak i na návrhu komunikace na globální úrovni.

8.1 Návrh vzdálené komunikace

Návrh se odvíjí od požadavků vyplývajících ze zadání diplomové práce. Proto musí komunikace splňovat možnost vzdáleného ovládání v sítích WAN a LAN. Na obr. 57 je naznačen základní princip pro vzdálené ovládání, který částečně vychází i z návrhu komunikace v bakalářské práci autora.



Obr. 57: Návrh vzdálené komunikace se zařízením.

Navržená komunikace počítá s využitím obslužného počítače, který je fyzicky propojen rozhraním USB s přípravkem. Komunikace na rozhraní USB je realizována prostřednictvím virtuálního COM portu a standardu pro sériovou komunikaci typu USART, která je i hardwarově podporována přímo mikrokontrolérem ATmega128. Počítač pak zajišťuje napojení do sítí LAN a WAN. Komunikace v sítích LAN je zajištěna protokolem TCP/IP. Komunikace v sítích WAN je řízena úpravou tabulek

v externí MySQL databázi jak ze strany ovladače spuštěném na obslužném počítači, tak i internetovým uživatelským systémem. Komunikace s databází probíhá standardními příkazy typu SQL. Díky předávání informací přes databázi není nutné instalovat podpůrný software na straně uživatele internetových stránek. Komunikace se tím značně odlišuje od běžného řešení konkurenčních laboratoří a umožňuje velice jednoduché ovládání z libovolného místa, kde je přístup na Internet a to včetně mobilních zařízení.

Celá komunikace se řídí pyramidovou hierarchií, na jejímž vrcholu je internetová aplikace a všechny prvky jsou ve svém chování vždy podřízeny svému nadřazenému členu. Nadřazený člen dokáže nižší prvek plně ovládat a je vždy informován o jeho aktuálním stavu a chybových hlášeních. Zpětná vazba při komunikaci je realizovaná formou textu, grafiky a videa, proto je uživatel dostatečně a kvalitně informován o aktuálním stavu i v případě vzdáleného řízení. Množství přenášených dat je pro jednotlivé úrovně komunikačních členů optimalizováno tak, aby bylo zajištěno ovládání a sběr dat v reálném čase. Komunikace přes Internet je dimenzována na přenos téměř 5000 hodnot za sekundu a uživatel proto není ochuzen o žádná podstatná data.

8.2 Embedded software

Jedná se o software navržený přímo pro mikrokontrolér ATmega128, který se skládá z velkého balíků funkcí pro zajištění funkčnosti všech mechanických prvků, měření se senzory a komunikaci s obslužným počítačem. Software je napsaný v jazyku C, který je optimalizovaný pro vývoj mikrokontrolérů firmy ATMEL ve formě knihovny AVR-GCC. Vývoj probíhal ve vývojovém prostředí AVR Studio 4.0 a díky nainstalovanému bootloaderu s podporou protokolu STK500 byl tímto prostředím mikrokontrolér i programován.

Celá struktura navrženého softwaru se dá rozdělit do několika celistvých bloků:

- míchání roztoků,
- měření s elektrochemickými senzory,
- kalibrace čerpání,
- stavový automat.

8.2.1 Míchání roztoků

Jedná se o softwarový blok zahrnující ovládání krokových motorů, pump a zpětnovazebních členů. Funkce ovládající krokové motory zahrnuje i jejich autokalibraci sledováním signálů z mikrospínačů umístěných na plastovém běžci. Funkce se stará o výpočet přesného počtu kroků pro nájezd na definované pozice zvolené uživatelem a postupný rozběh na maximální rychlost pohybu běžce. Ovládání pump spočívá v definování směru, doby a režimu čerpání a samozřejmě ve volbě mezi

„pomalou“ a „rychlou“ pumpou. Komunikace se zpětnovazebními členy čerpání roztoků spočívá v zapnutí senzorů a vyhodnocení signálů ze čtveřice komparátorů. Důsledkem komunikace je zajištění správného odkapávání kapek roztoků z přečerpávacího hrotu a informování uživatele o správné funkčnosti.

8.2.2 Měření s elektrochemickými roztoky

Stejně jako jedním z nejdůležitějších hardwarových bloků byla jednotka potencisotatu tak i tento softwarový blok má stejnou prioritu na úrovni softwaru. Slouží pro nastavení rozsahu měřeného proudu, zajišťuje definování napěťových úrovní přiváděných na čip IMAM dle zvolené měřicí metody a snímání hodnoty proudu v okamžicích určených danou metodou. Tento blok zajišťuje i volbu měřicího kanálu, průměrování naměřených hodnot proudu a přípravu dat pro odeslání záznamu měření do obslužného počítače.

Blok implementuje do zařízení několik typů měření na základě vybraných voltametrických metod, které jsou vhodné pro digitální zpracování:

- stejnosměrná voltametrie,
- cyklická voltametrie,
- schodovitá voltametrie,
- square-wave voltametrie,
- pulzní polarografie,
- diferenciální pulzní polarografie,
- potenciometrie.

8.2.3 Kalibrace čerpání

Blok pro kalibraci čerpání obsahuje řadu velmi důležitých funkcí pro správné nastavení doby čerpání definovaného množství roztoků pro jednotlivé čerpací rychlosti:

- $1 \mu\text{l} \cdot \text{min}^{-1}$,
- $10 \mu\text{l} \cdot \text{min}^{-1}$,
- $100 \mu\text{l} \cdot \text{min}^{-1}$,
- $1 \text{ ml} \cdot \text{min}^{-1}$,
- $10 \text{ ml} \cdot \text{min}^{-1}$.

8.2.4 Stavový automat

O zastřešení kooperace jednotlivých funkcí a softwarových bloků na základě požadavků uživatele se stará blok typu stavový automat. Řídí spouštění jednotlivých funkcí v definovaném pořadí nebo čase. Ovládá i funkční bloky pro komunikaci

s obslužným počítačem, zobrazení informací na LCD displeji a stará se o inicializaci po zastavení činnosti způsobené stisknutím stop tlačítka na čelním panelu.

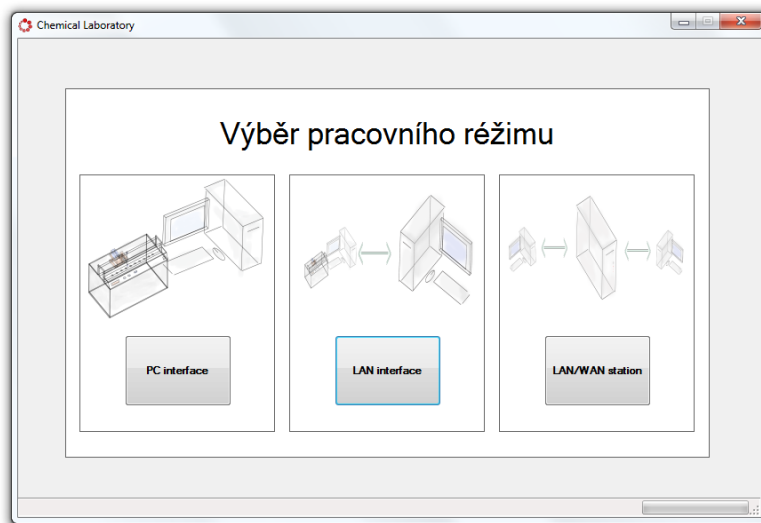
8.3 Ovladač pro obslužný počítač

Ovladač pro obslužný počítač je aplikace pro systém Windows naprogramovaná v jazyce C#. Volba programovacího jazyku C# vyplývá z toho, že v něm jsou implementovány knihovny pro komunikaci se sériovým portem a lze pro něj nalézt bohatou podporu ze strany vývojářů a řadu knihoven pro splnění rozmanitých úkolů. Navíc s využitím vhodného softwaru, jako je Visual Studio od společnosti Microsoft, je tvorba aplikace pro systém Windows poměrně snadná a rychlá.

Ovladač je vyšší softwarovou vrstvou v hierarchii vzdáleného ovládání pracoviště elektrochemických senzorů. Aplikace může pracovat ve třech režimech:

- PC interface,
- LAN interface,
- LAN/WAN station.

Při spuštění je uživateli umožněn výběr režimu (obr. 58), ale po uplynutí doby 15 sekund je aplikace automaticky spuštěna jako LAN/WAN station. Automatické spuštění se děje pouze v případě, že uživatel nevybral žádnou volbu. Tato akce se provádí vzhledem k primárnímu účelu ovladače vytvářet komunikační uzel mezi přípravkem a vzdáleným uživatelským prostředím.



Obr. 58: Ovladač – výběr pracovního režimu.

Ovladač obsahuje celou řadu nejrůznějších funkcí, k nimž má uživatel přístup při volbě jednotlivých režimů. Funkce jsou zpřístupněny podle jejich charakteru, tedy jestli souvisejí s přípravkem nebo s komunikací. Proto například formulář pro kalibraci čerpání nelze spustit na vzdáleném pracovišti. Mezi samozřejmost patří export

naměřených dat ve formě, která je vhodná pro následné zpracování např. v Excelu. Nastavení komunikace s databází vychází z defaultní předvolby hned po nainstalování, ale v případě potřeby ji lze změnit. Rovněž komunikace s přípravkem je navržena tak, aby byl automaticky identifikován komunikační port, ale lze jej také manuálně změnit.

Každá sekce ovládání se řídí pravidly pro možnosti tlačítek START, STOP a PAUSE. Po spuštění tlačítkem start se provede ověření spojení se všemi úrovněmi komunikace v dané komunikační cestě. Může se tedy provést ověření spojení s přípravkem, databází, komunikačním uzlem případně i videokamerou. Tlačítko stop vždy ukončí aktuální činnost a tlačítko pause ji naopak pozdrží. O všech chybách a stavech je uživatel průběžně informován.

Kooperace mezi jednotlivými funkcemi, vlákny, režimy a všemi prováděnými akcemi je řízena stavovým automatem obdobnou formou jako v softwarovém vybavení mikrokontroléru ATmega128 navrženého zařízení.

Ovladač nevyžaduje žádné speciální zacházení, rovněž instalace je automatická a bezproblémová. Pokud na hostujícím počítači chybí potřebný software, jako např. .NET framework, je v instalačním balíčku vyžadovaný software zahrnut. Po nainstalování se ovladač objeví v nabídce Start a může být pouhým přesunem do složky „Po spuštění“ automaticky volán hned při startu systému. Vývojové prostředí Visual Studio umožňuje publikovat aplikaci i v rámci Internetu, což se dá s výhodou využít pro automatickou detekci nových verzí softwaru. Takto publikovaná aplikace poté před každým spuštěním upozorní uživatele na dostupnost nové verze v případě, že byl software upraven.

8.3.1 PC interface a LAN interface

Vzhled a možnosti interfacu nebo-li uživatelského prostředí typu PC nebo LAN, se v zásadě neliší. Možnost výběru je ovšem zásadní pro typ komunikace, množství přenášených dat a aktivaci dostupných nastavení. Rozdíly vycházejí z toho, že PC interface je primárně určeno pro ovládání zařízení přímo na obslužném počítači a LAN interface slouží pro vzdálené ovládání z počítače připojeného do vnitřních sítí typu LAN. Interfacem lze nastavit volbu roztoků a provádět měření vybranou měřicí metodou.

Při spuštění se uživateli zobrazí tři záložky a menu pro nastavení a export dat. Každá záložka je specifická jak souborem funkcí, tak i chováním a umožní volbu mezi:

- mícháním roztoků,
- měřením se senzorem a
- získáním signálu z videokamery.

8.3.2 Míchání roztoků

Míchání roztoků (obr. 59) je optimalizováno pro jednoduchost ovládání a názornost konkrétní činnosti uživatele. Skladbu roztoku je možné tvořit jak využitím funkčních formulářových polí a tlačítek nebo formou editace textového výstupu. Zpětnou vazbou pro libovolný způsob vytvoření roztoku je úprava hodnot progressbarů znázorňujících obsah ampulí a nádob.

Uživatel si nejdříve nasimuluje postup namíchání roztoku a po stisknutí tlačítka start zahájí jeho reálné míchání. Hodnoty obsahu jsou při načerpání roztoku upraveny odepsáním načerpaného množství od aktuálního objemu. Skladba roztoku v reakční nádobě se upravuje až po vyčerpání do prostoru nádoby. Upravené stavy jsou zaznamenány do paměti aplikace, aby uživatele vždy aktuálně informovaly o stavu roztoku a dostupných objemových množstvích složek pro míchání.

Doplnění ampulí a nádob se děje přes speciální formulář určený pro doplňování stavů roztoků. Pole formulářových polí obsahují také možnost pro pojmenování pozic P1 – P8 a identifikaci speciálních znaků pro automatické vyprázdnění nebo dopočet množství do plného stavu.

V režimu PC interface je možné spustit i formulář pro kalibraci rychlosti čerpání „rychlé“ a „pomalé“ pumpy. Jedná se o funkci sledující čas vyčerpání definovaného množství v závislosti na zpětné vazbě s obsluhou.

Chemical Laboratory - PC Interface

Nastavení EXPORT dat

Míchání roztoku Měření senzoru Kamerový systém

P1	P2	P3	P4	P5	P6	P7	P8	Pufr	Oplach
peroxid 30	vzorek_2	vzorek_3	vzorek_4	vzorek_5	vzorek_6	vzorek_7	vzorek_8	300 ml	500 ml
14975 ul	0 ul	0 ul	0 ul	0 ul	0 ul	0 ul	0 ul	300 ml	500 ml
zvolit	zvolit	zvolit	zvolit	zvolit	zvolit	zvolit	zvolit	zvolit	zvolit

Zvoleny roztok: P1

Mnozství: ul

Nasobek: krat

Obsah kadinky se senzorem:

Pufr 200ml

P1 25ul

Kadinka se senzorem

Obr. 59: Ovladač – míchání roztoků.

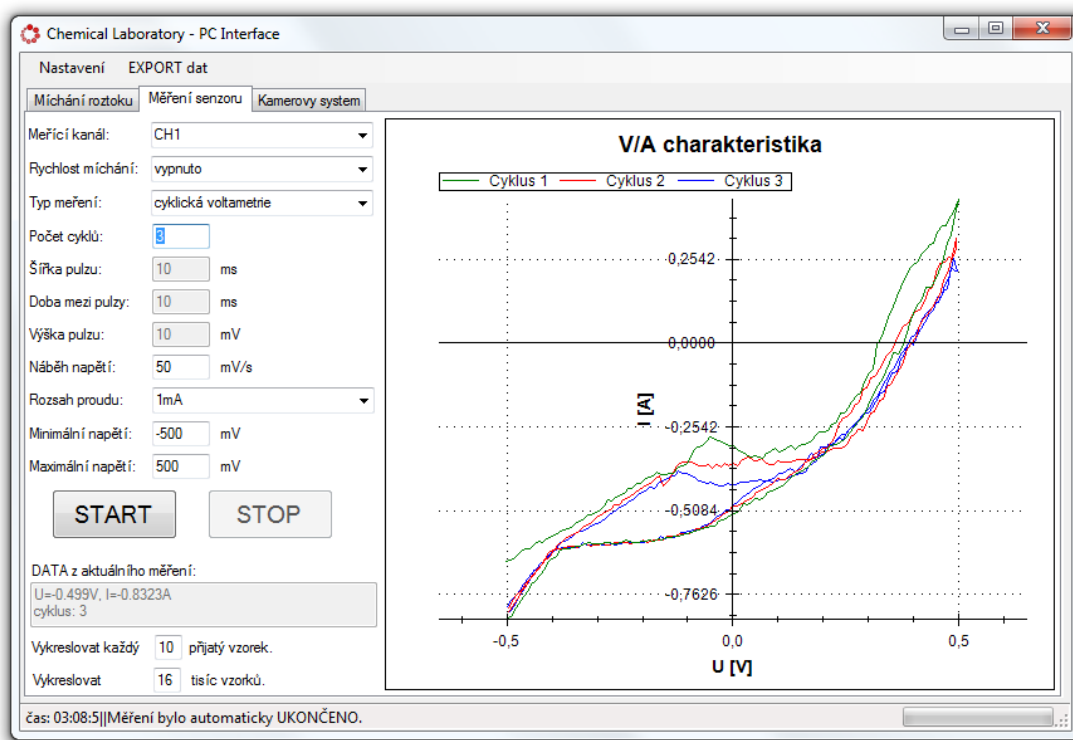
8.3.3 Měření se senzorem

Druhá záložka uživatelského prostředí ovladače se věnuje ovládání jednotky potenciostatu a sběru dat při testování senzorů. Jak již z pořadí záložek vyplývá, aby bylo možné spustit měření, musí být obsah reakční nádoby nenulový. Nastavení měření spočívá v pěti krocích:

- volba měřicího kanálu,
- rychlost míchání,
- typ měření,
- parametry měření,
- rozsah proudu a napětí.

Doplňující nastavení souvisejí s vykreslováním grafu, aby jeho zobrazení bylo reálné a dostatečně pružné. Nutnost nastavení parametrů vykreslování vychází z požadavku o bezchybný příjem všech dat z měření. Přílišné zahlcení grafu daty může způsobovat nežádoucí mrznutí nebo zpoždování aplikace, které má vliv i na asynchronní programová vlákna, která jsou aplikací spuštěná.

Jednotlivé parametry měření jsou aktivovány nebo deaktivovány dle zvolené měřicí metody a jejich nastavením se dá docílit průběhů, které vycházejí z teoretické části práce věnující se elektrochemickým metodám.



Obr. 60: Ovladač – měření senzoru.

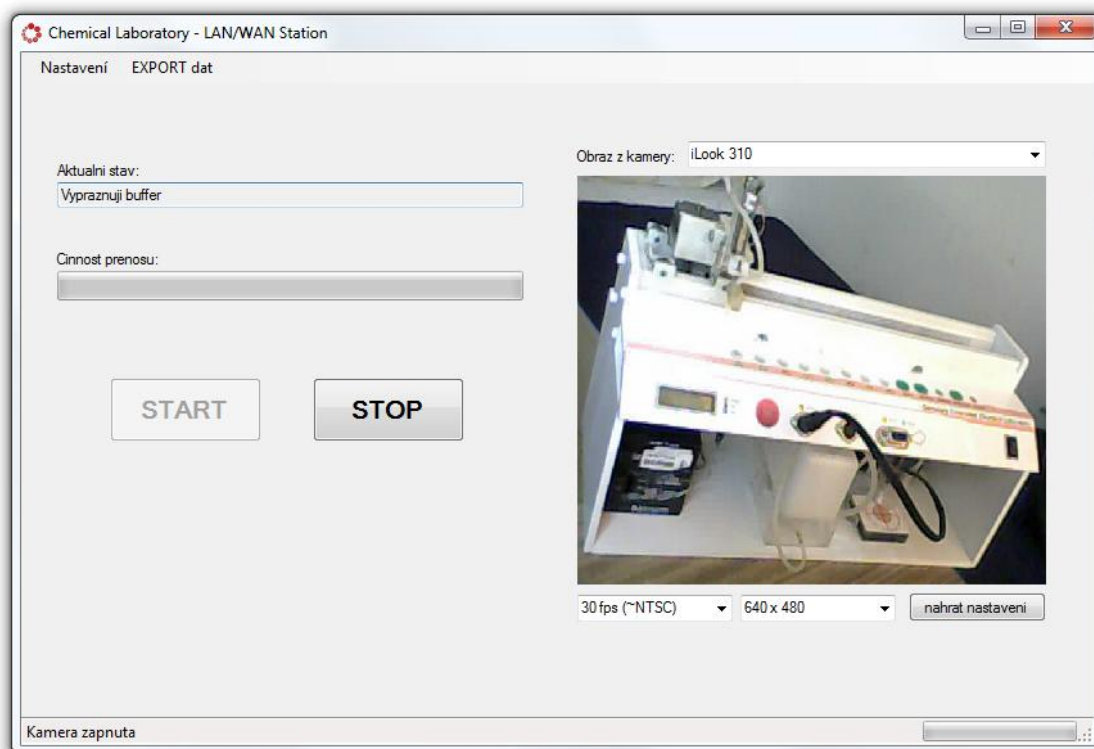
8.3.4 Kamerový systém

Záložka kamerového systému slouží jako zpětná vazba hlavně pro uživatele vzdáleného uživatelského prostředí v sítích LAN. Nastavení generovaného videa je možné pouze v režimu stanice spuštěné na obslužném počítači.

8.3.5 LAN/WAN station

LAN/WAN station nebo-li LAN/WAN stanice je režim do nějž musí být ovladač překlopen na obslužném počítači vždy, když je potřebné získat přístup k zařízení formou vzdáleného ovládání. Stanice funguje jako komunikační uzel, který obsluhuje současně síť WAN (Internet) i LAN (lokální síť). Ovládání je, ale vždy možné pouze z jediného PC umístěného v některé ze sítí. Ostatní připojené PC se chovají pouze jako účastníci měření a nemohou do něj žádným způsobem zasahovat, pokud právě probíhá. Výhodou účastnické realizace je možnost měření skupiny lidí, kdy jeden uživatel je v roli operátora a ostatní uživatelé měření pouze sledují. Ovšem v případě návrhu změn parametrů měření může roli operátora po domluvě převzít, kterýkoliv jiný uživatel.

Formulář stanice je jednoduchý a zobrazuje pouze vyprazdňování komunikačního bufferu (zásobníku), textový výpis aktuálního prováděného stavu a obraz z videokamery. V režimu stanice je zpřístupněn výběr videokamery, určení formátu videosignálu a v nastavení změna spojení s databází. V neposlední řadě lze získat informace o parametrech pro navázání spojení v sítích LAN, které se musí zadat v nastavení přístupném z režimu LAN interface.



Obr. 61: Ovladač – LAN/WAN station.

8.4 Internetová aplikace

Internetová aplikace, na obr. 62, je založená na programovacím jazyku PHP a speciálním použití Java Scriptu, který napodobuje chování AJAXu tím, že vykonává potřebné instrukce na pozadí. Kombinace těchto technik umožňuje vykreslovat grafy VA charakteristik on-line v reálném čase a navíc bez nutnosti aktualizace internetového prohlížeče. Prohlížeč se nemusí aktualizovat, protože se část skriptu zpracuje na uživatelské straně a není nutné čekat na zpracování serverem.

Prvním důležitým krokem pro přístup do uživatelského prostředí internetové aplikace je přihlášení uživatele (obr. 62). Přihlášení je bezpodmínečným krokem hlavně z důvodů ochrany jak navrženého zařízení, tak i průběhu měření. V případě LAN interfacu je identifikace zaznamenána přesným určením stroje, z něž bylo navázáno spojení a to získáním IP adresy a MAC adresy spolu s časovým údajem spojení. Takový způsob ochrany u internetových aplikací nelze dosáhnout, protože:

- PHP ani žádný jiný programovací jazyk pro tvorbu internetových stránek není schopen získat IP adresu a MAC adresu stroje připojeného uživatele,
- lze získat pouze IP adresu a MAC adresu serveru, z něž se uživatel připojuje.

Získané informace o serveru nejsou dostatečně prokazatelné pro konkrétního uživatele, protože z jednoho serveru se může připojovat velké množství uživatelů a všichni proto budou mít IP a MAC adresu stejnou. Jediný způsob jak zajistit identifikaci uživatele je jeho přihlášením a získáním doplňujících informací v podobě IP adresy a času připojení.

Aplikace se poté chová stejně jako LAN interface v ovladači a umožní ovládat přípravek pouze z jednoho počítače jediným uživatelem. Ostatní uživatelé mohou měření pouze sledovat. Konflikt přístupu je zajištěn spuštěním prvního zadaného řetězce příkazů s dostatečně přesným časovým údajem o zápisu do fronty příkazů. Ostatní uživatelé musí počkat na ukončení aktuálního měření.

Ochrana přístupu je navíc zvýšena tím, že se nelze do aplikace zaregistrovat běžným způsobem. Registraci nových uživatelů provádí vždy uživatel s dostatečnou úrovní pravomocí po svém přihlášení v příslušné nabídce pro registraci nových uživatelů. Tento způsob ochrany vede sice k zásadnímu odepření přístupu k navržené aplikaci široké veřejnosti, ale návrh ani zadání s touto možností nepočítalo. Přístup se uděluje spíše uživatelům na akademické půdě či jiným zainteresovaným uživatelům.

RC Chemical Lab

WEB INTERFACE

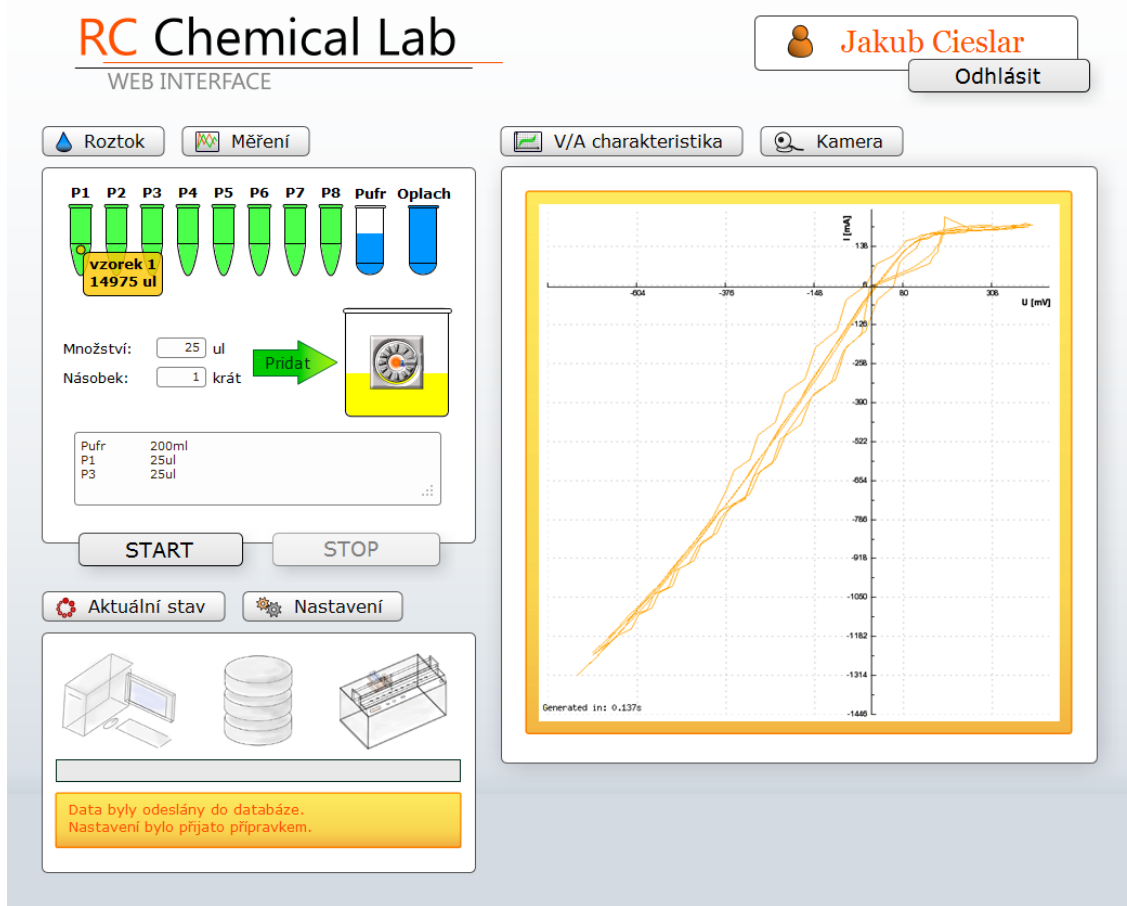


 **Přihlášení do aplikace**

Obr. 62: Internetová aplikace – přihlášení.

Funkčnost aplikace je naprosto stejná jako interface ovladače pro obslužný počítač, a proto zde již nebude popisována. Jediným rozdílem je značně vyšší uživatelská přívětivost (obr. 63) založena na důrazu grafické interpretace jednotlivých funkcí. K přehlednosti přispívá i kompaktní řešení jednotlivých bloků a grafické značení pomocí ikon. Uživatelé mohou volit mezi výstupem z kamery nebo zobrazením grafu. Pro dosažení dostatečné rychlosti komunikace se dají nastavit parametry pro vykreslování grafu, ale o naměřené hodnoty uživatel není ochuzen. Při ukončení měření je uživatel vždy dotázán na možnost získání veškerých naměřených dat formou exportu do databáze. S daty v databázi pak může uživatel nakládat dle libosti ať už stažením souboru s exportovanými daty nebo zobrazením dat přímo v aplikaci formou postprocessingu.

Pro větší efektivitu přenosu dat byl vytvořen i protokol pro přenos videosignálu formou databázových tabulek. Výhodou tohoto řešení je nižší zatížení serveru využíváním komunikace pouze s databází. Lze lépe řídit funkci přenosu a zabráňuje se odesílání dat, která nejsou prohlížena. Volba mezi zobrazením grafu nebo videa byla vytvořena záměrně, aby nebyl současně přijímán příliš velký balík dat. Uživateli příliš nebude vadit nemožnost současného pozorování vykreslování grafu a videa, protože při měření se na přístroji neděje nic zásadního, co by vyžadovalo větší pozornost. Naproti tomu při míchání roztoků je video potřeba a data pro graf nejsou vytvářena vůbec, takže je není nutné ani sledovat. Uživatel si ovšem může typ vykreslovaných data libovolně a kdykoliv zvolit.



Obr. 63: Internetová aplikace – uživatelské prostředí.

Blok, který neustále informuje uživatele o aktuálním dění je ve spodní části aplikace. Je na něm zobrazeno vyprazdňování komunikačního bufferu mezi Internetovou aplikací a databází a mezi přípravkem a databází. Zároveň jsou vypisovány i textové zprávy a dochází k aktualizaci objemových stavů ampulí, nádob a reakční nádoby. V horní části aplikace je možnost pro přístup do správy uživatelského účtu a registraci nových uživatelů a také možnost odhlášení uživatele.

Aplikace má ošetřený i neautorizovaný přístup k programovým blokům, které jsou využívány jako zásuvné moduly. Hlášení o chybách ve spojení s databází je samozřejmostí. Sekce věnující se aktuálnímu stavu přípravku sice zobrazuje veškerá chybová hlášení, ale textová forma výpisu ve spodní části okna může být uživateli snadně přehlednější. Proto závažný typ hlášení jako např. chyby ve spojení na komunikační cestě, pád aplikace na obslužném počítači, obsazenost zařízení jiným uživatelem a jiné chybové stavy, které zabraňují správnému ovládání nebo komunikaci, jsou vypisovány ve speciální podobě v horní části aplikace nad ovládacími bloky.

9 Měření

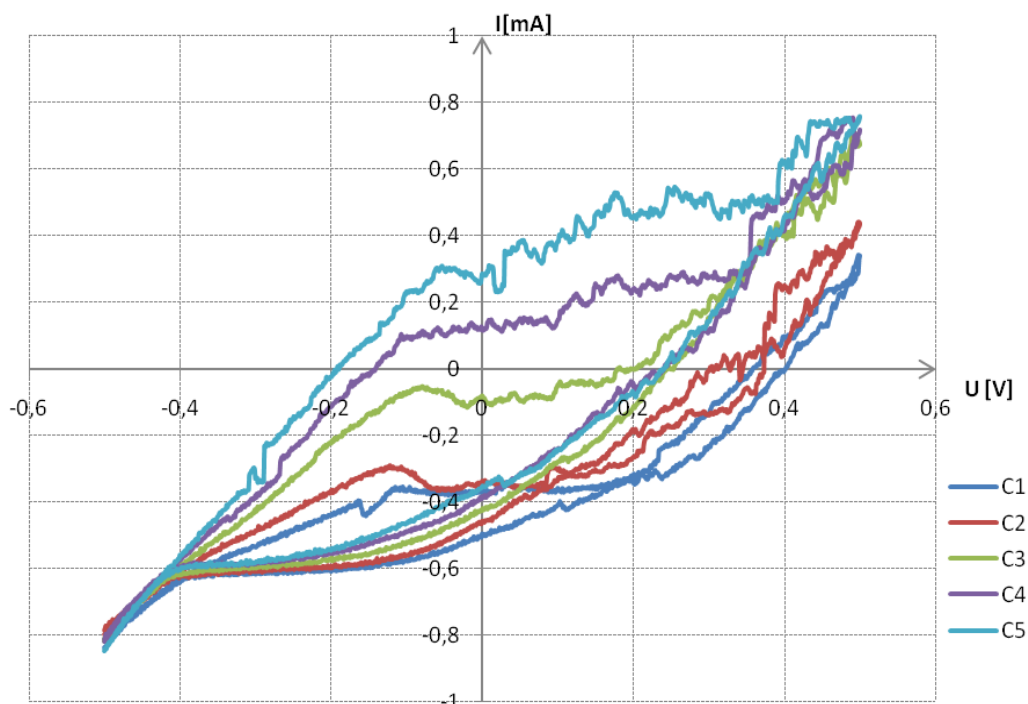
Měření probíhalo v několika fázích a jeho výsledky značným způsobem ovlivňovaly vývoj jak zařízení, tak softwarového vybavení. Je nutné podotknout, že testovací fáze není ani z daleka u konce a je nutné na základě výsledků provést nutné kroky k nápravě nedostatků.

Návrh mechanického řešení zařízení se po pár drobných úpravách osvědčil a funguje dle předpokladů, nicméně nejsou ověřeny některé použité materiály.

Správná funkce IMAM, ovládacího zařízení i obslužného softwaru byla ověřována na pokusném zapojení obsahující nelineární prvek. Provedlo se i dodatečné proměření kontrolního roztoku, ale vývoj pokračoval s nelineárním prvkem z důvodu agresivity a nebezpečnosti používaných chemikálií.

Měření kontrolního roztoku je dokumentováno na obr. 64. Provedla se cyklická voltametrie pro roztok s borátovým pufrům a zvyšující se koncentrací 30 % peroxidu vodíku. Výsledky měření byly vyexportovány a zpracovány Excelem pro větší přehlednost. Z grafu je patrný růst píku vyznačující koncentraci peroxidu až do hranice nasycení měřeného roztoku.

V/A charakteristika - Borátový pufr a rostoucí koncentrace peroxidu vodíku (30 %)



Obr. 64: V/A charakteristika kontrolního roztoku.

10 Závěr

Diplomová práce je zaměřena na návrh a realizaci zařízení pro automatizované měření s tlustovrstvými elektrochemickými senzory, které je možné vzdáleně ovládat. Vzdáleným ovládáním je umožněn efektivnější vývoj tlustovrstvých senzorů v rámci jejich testování. Zvolený přístup řešení internetové komunikace lze považovat za inovativní, protože navržený způsob nevyžaduje instalaci serveru na straně zařízení ani komunikačního softwaru na straně uživatele internetového rozhraní.

Úkolem diplomové práce bylo seznámit se základními měřicími metodami v oblasti elektrochemických senzorů. Seznámit se s jazyky pro tvorbu zabudovaného programového vybavení (embedded software), aplikací pro operační systém Windows a aplikací pro tvorbu internetového rozhraní (např. C, PHP, HTML). Na základě nabytých teoretických znalostí byl zahájen návrh řešení dálkově řízeného pracoviště elektrochemických senzorů, které bude řízeno v rámci sítí WAN, případně LAN.

První část návrhu se věnovala mechanické konstrukci zařízení na základě poznatků získaných při současném manuálním způsobu měření. Z těchto poznatků byl sestaven postup měření a základní parametry, které navržené zařízení musí splňovat. Dodržením požadovaných parametrů byl docílen návrh v podobě 3-D modelu, který se použil jako předloha pro výrobu základních konstrukčních prvků. Konstrukční prvky byly poté upraveny na základě potřeb pro správnou funkčnost reálného zařízení a splnění požadavků vycházejících z praktického používání zařízení.

Druhá část návrhu se věnovala obvodovým blokům, které slouží pro ovládání mechanických prvků a měření s tlustovrstvými elektrochemickými senzory. Navržené obvody byly rozděleny do čtyř bloků, které je možné dle potřeb libovolně upravovat. Tím je zajištěna i možnost snadného dalšího vývoje zařízení na základě poznatků z praxe.

Třetí část návrhu se věnovala vývoji softwaru. Software byl vyvíjen na třech úrovních a to software pro mikrokontrolér, obslužný počítač a síť LAN a internetové uživatelské prostředí. Byla zajištěna pyramidová komunikace mezi jednotlivými úrovněmi a funkčnost byla ověřena praktickou zkouškou vzdáleného řízení.

Funkčnost realizovaného zařízení jako celku byla ověřena reálným měřením. Měření potvrdilo správný návrh jednotky potenciostatu i mechanické konstrukce zařízení. Rovněž se ověřil návrh komunikace včetně přenosu videosignálu a naměřených dat v sítích LAN a WAN.

Konečným výsledkem diplomové práce je zařízení, které je možné ovládat jak obslužným počítačem, tak vzdáleně přes Internet a síť LAN. Zařízení umožňuje

automaticky namíchat roztok do reakční nádoby a provést měření se senzorem vybranou měřicí metodou.

Navržené zařízení rovněž umožňuje i široký prostor pro další vývoj a zlepšení parametru měření zaměřením se na:

- volbu vhodných materiálů pro expanzní nádoby a průchodky (sklo),
- úpravu jednotky potenciostatu pro rychlejší sběr dat, využitím programovatelného logického obvodu typu FPGA nebo CPLD a vhodné analogové úpravy signálů,
- vývoj nových i stávajících metod měření a
- vývoj softwarového vybavení jako celku.

Seznam použitých zdrojů

- [1] Vysoké učení technické v Brně: *Moodle*. [online]. [citováno dne 2009-05-18]. Dostupné z: <http://moodle.fce.vutbr.cz/>.
- [2] National Instrumensts: *NI LabVIEW*. [online]. [citováno dne 2009-05-18]. Dostupné z: <http://www.ni.com/labview/>.
- [3] ISES WEB Control: *Softwarová stavebnice "ISES WEB Control" pro jednoduchou tvorbu vzdálených experimentů*. [online]. [citováno dne 2009-05-18]. Dostupné z: http://www.ises.info/old-site/Lustig_ICTE_c.html.
- [4] Univerzita Karlova v Praze, Matematicko-fyzikální fakulta: *Řízení výšky vodní hladiny*. [online]. [citováno dne 2009-05-18]. Dostupné z: <http://kdt-14.karlov.mff.cuni.cz/cz/mereni.html>.
- [5] CIESLAR, J. *Pracoviště senzorové techniky přístupné pomocí internetového rozhraní*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 55 s. Vedoucí bakalářské práce Ing. Martin Adámek, Ph.D.
- [6] Hubálek, J., Adámek, M.: *Mikrosenzory a mikroelektromechanické systémy*. Brno: Vysoké učení technické v Brně. Interní učební texty. VUT v Brně, 2007.
- [7] MACHÁŇ, L. *Detekce přítomnosti olova v pájkách používaných v elektrotechnice*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 60 s. Vedoucí diplomové práce Ing. Martin Adámek, Ph.D.
- [8] SZEINDUCH, Ivan. *Základy technologie mikroelektronických obvodů a systémů. 1. vydání*. Brno: VUTIUM, 2006. 375 s. ISBN 80-214-3292-6.
- [9] ADÁMEK, M., ŘEZNÍČEK, M., PRÁŠEK, J. *Moderní technologie elektronických obvodů a systémů – laboratorní cvičení nekonvenční aplikace tlustých vrstev a její využití v senzorové technice senzory*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010, 16 s.
- [10] Wikipedia.org: *Apache HTTP Server*. [online]. [citováno dne 2011-05-18]. Dostupné z: http://cs.wikipedia.org/wiki/Apache_HTTP_Server.
- [11] Wikipedia.org: *MySQL*. [online]. [citováno dne 2011-05-18]. Dostupné z: <http://cs.wikipedia.org/wiki/Mysql>.
- [12] Wikipedia.org: *HyperText Markup Language*. [online]. [citováno dne 2011-05-18]. Dostupné z: <http://cs.wikipedia.org/wiki/Html>.

- [13] Wikipedia.org: *JavaScript*. [online]. [citováno dne 2011-05-18]. Dostupné z: <http://cs.wikipedia.org/wiki/JavaScript>.
- [14] Wikipedia.org: *PHP*. [online]. [citováno dne 2011-05-18]. Dostupné z: <http://cs.wikipedia.org/wiki/Php>.
- [15] Root.cz: *PHP: úvod*. [online]. [citováno dne 2011-05-18]. Dostupné z: <http://www.root.cz/clanky/php-uvod/>.
- [16] Wikipedia.org: *C#*. [online]. [citováno dne 2011-05-18]. Dostupné z: http://cs.wikipedia.org/wiki/C_Sharp.
- [17] Wikipedia.org: *C++*. [online]. [citováno dne 2011-05-18]. Dostupné z: <http://cs.wikipedia.org/wiki/C%2B%2B>.
- [18] Wikipedia.org: *Jazyk symbolických adres*. [online]. [citováno dne 2011-05-18]. Dostupné z: <http://cs.wikipedia.org/wiki/Jsa>.
- [19] Atmel. Datasheet: *ATMega128*. [online]. [citováno dne 2011-05-18]. Dostupné z: <http://www.alldatasheet.com/datasheet-pdf/pdf/313342/ATMEL/ATmega128.html>.
- [20] DŘÍNEK, M.: *AVR – opravdový RISC s FLASH pamětí*. [online]. [citováno dne 2011-05-18]. Dostupné z: http://avr.hw.cz/architektura/arch_avr.html.
- [21] FTDI Chip. Datasheet: *FT232BM*. [online]. [citováno dne 2011-05-18]. Dostupné z: <http://www.alldatasheet.com/datasheet-pdf/pdf/202272/FTDI/FT232BM.html>.
- [22] Atmel Products. Tools & Software: *AVR Studio*. [online]. [citováno dne 2011-05-18]. Dostupné z: http://www.atmel.com/dyn/products/tools.asp?family_id=607.
- [23] PonyProg: *Serial device programmer*. [online]. [citováno dne 2011-05-18]. Dostupné z: <http://www.lancos.com/prog.html>
- [24] Atmel. Datasheet: *ATTiny13-20PU*. [online]. [citováno dne 2011-05-18]. Dostupné z: <http://www.alldatasheet.com/datasheet-pdf/pdf/255866/ATMEL/ATTINY13-20PU.html>
- [25] VRBA, R., ŠTEFFAN, P., HÁZE, J., FUJCIK, L., SKOČDOPOLE, M., PROKOP, R. *IMAM Inteligentní mikrosenzory a mikrosystémy pro měření, regulaci a životní prostředí*. GAČR. Brno, Praha: GAČR, 2004. s. 0-55.

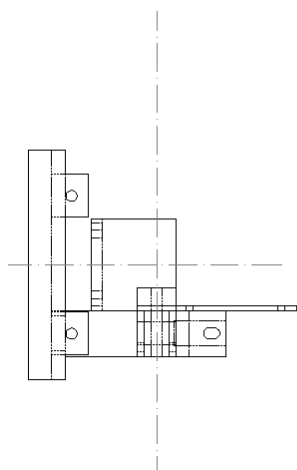
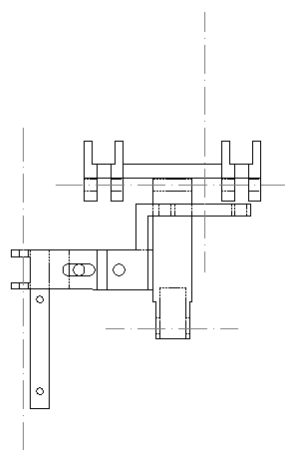
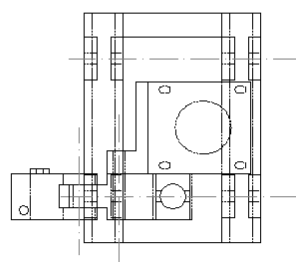
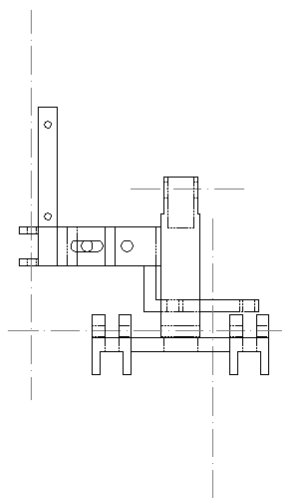
Přílohy

Příloha 1 – přiložené CD

CD obsahuje elektronickou verzi diplomové práce, data z měření, návody, manuály, instalační soubory, data pro návrhový systém Eagle 5.7.0, poslední verzi zdrojového kódu softwaru mikrokontroléru, ovladače pro obslužný počítač a internetového rozhraní a zálohu MySql databáze.

Příloha 2 – sestava běžce

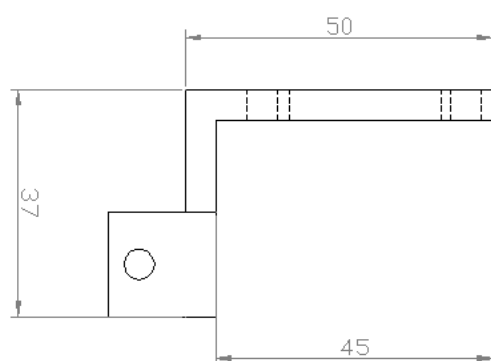
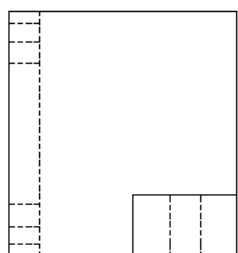
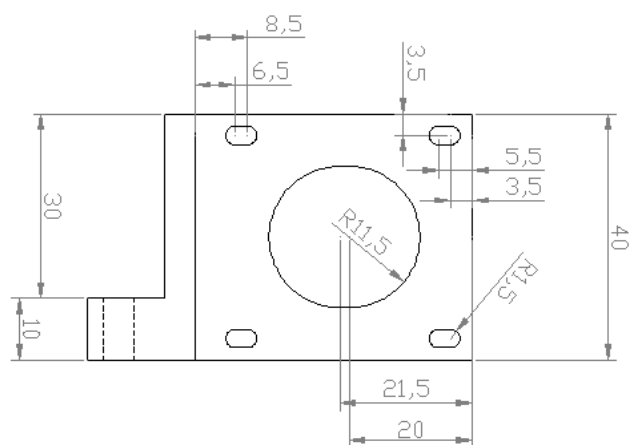
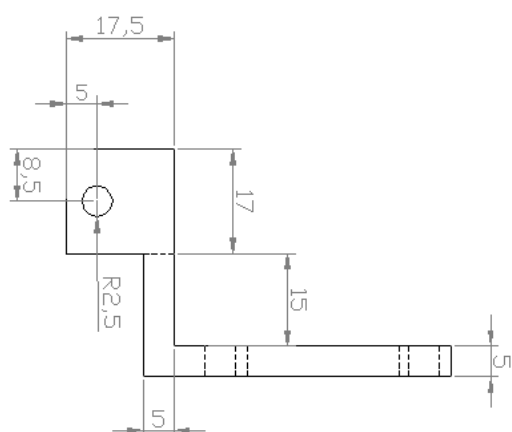
Sestava - Běžec

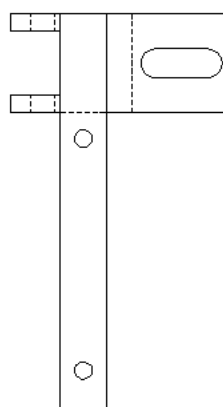
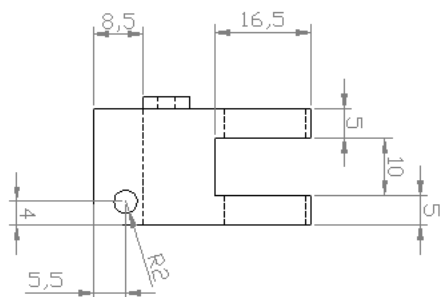
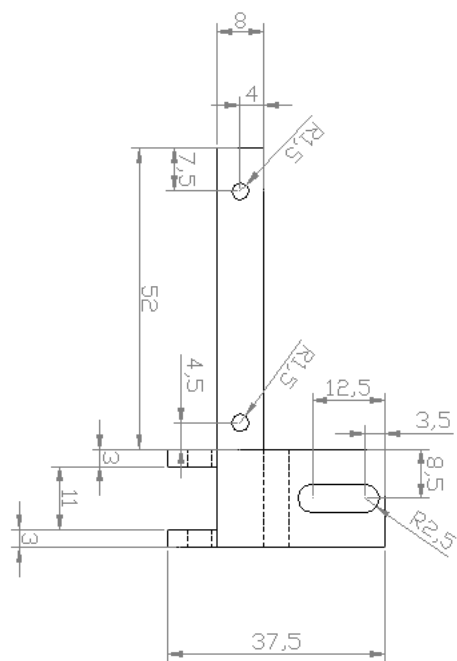


Kus č.: 1
Běžec - základna
Počet kusů: 1

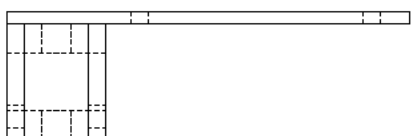


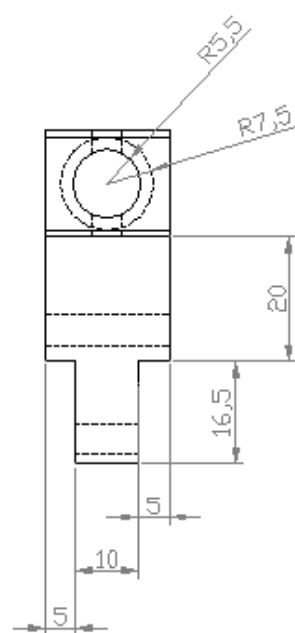
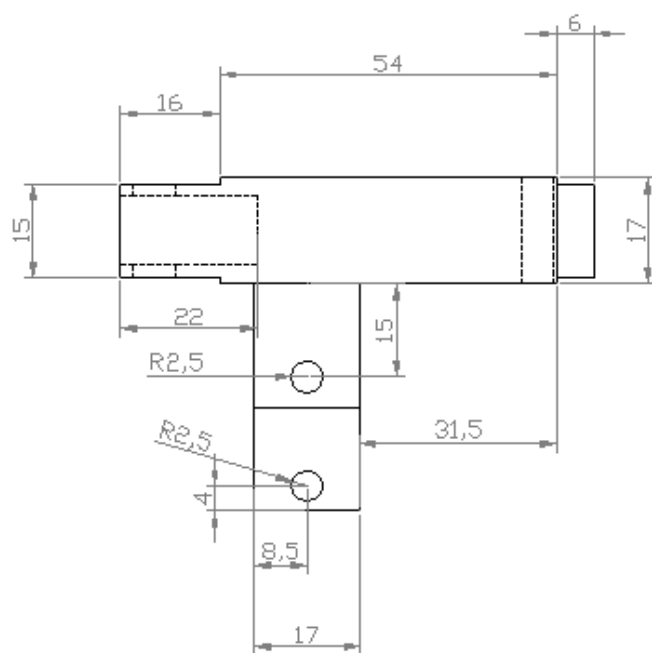
Kus č.: 2
Běžec - uložení motoru 1
Počet kusů: 1



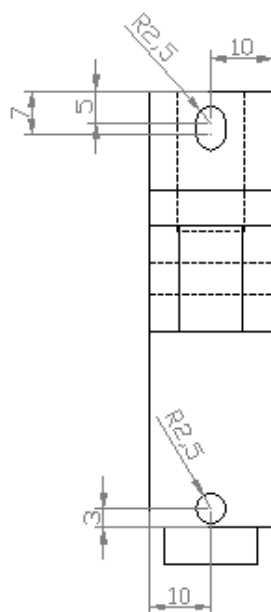


Kus č.: 3
Běžec - uložení motoru 2
Počet kusů: 1

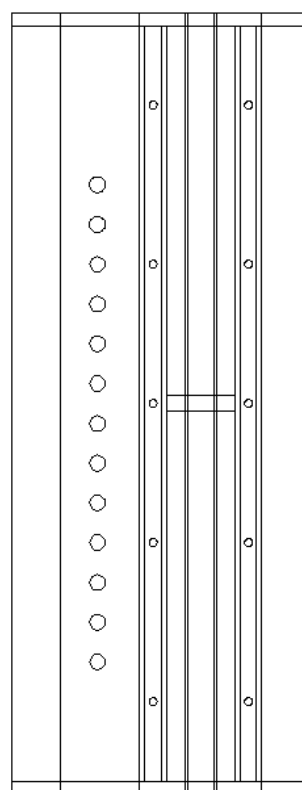
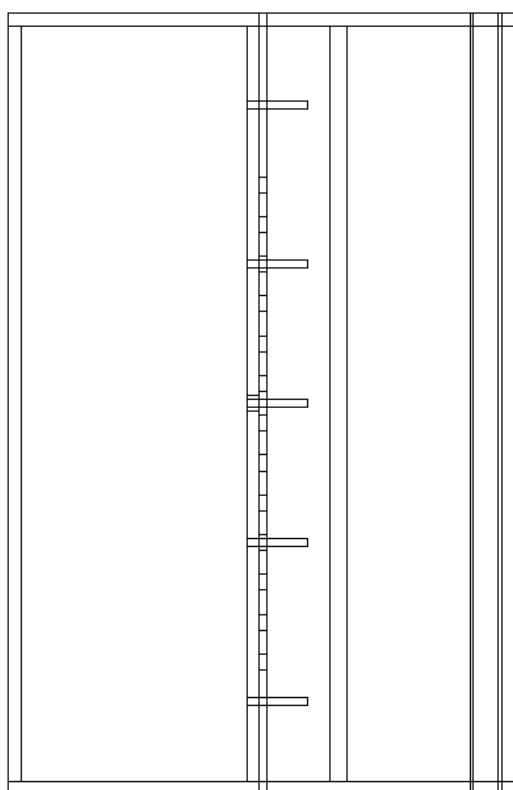




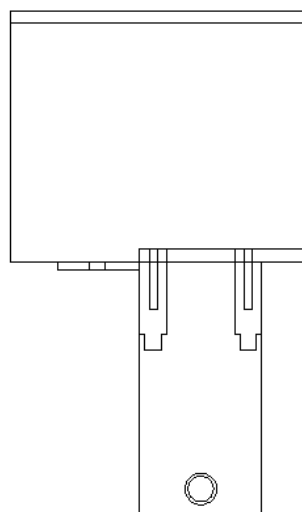
Kus č.: 4
 Běžec - středový sloup
 Počet kusů: 1



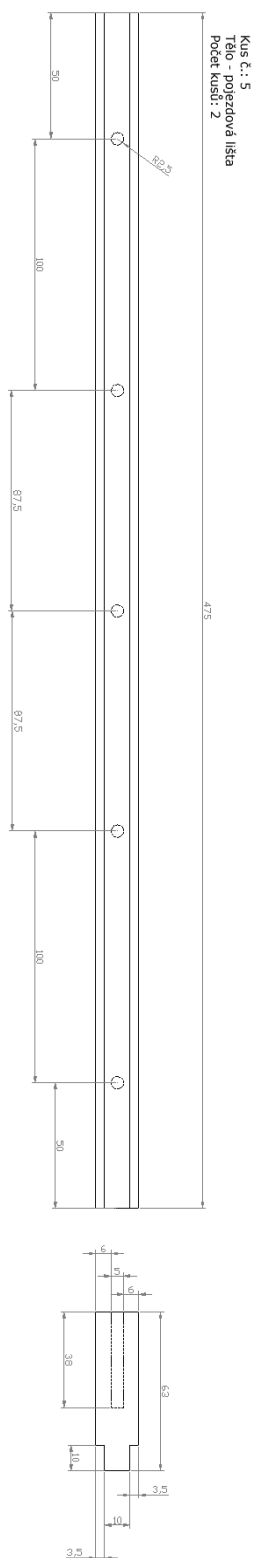
Příloha 4 – sestava těla zařízení



Sestava - Tělo



Příloha 5 – kusovníky pro výrobu těla zařízení



Kus č.: 6
Tělo - zásobník na zkumavky
Počet kusů: 1

