



Středoškolská technika 2016

Setkání a prezentace prací středoškolských studentů na ČVUT

Automatický akviziční systém pro pH a EC s datalogem

Michal Šrutka

Střední průmyslová škola Brno, Purkyňova, příspěvková organizace
Purkyňova 97, Brno

Anotace

Práce je zaměřena na sestavení zařízení, které umožňuje měření pH a EC u roztoků. Zabývá se obecnou konstrukcí měřících sond a elektroniky pro tyto sondy. V další části práce je sepsána realizace dané elektroniky pro dané sondy. Nakonec pak i realizace akvizičního FW a SW, který umožní hodnoty uložit v počítači.

Klíčová slova

akviziční jednotka; pH; EC; roztok; UART, USB, C#

Annotation

This thesis deals with design of measurement device for pH and EC. In first part of thesis there are description of kind of measurement technique. Next part of thesis contains design of my own device for this measurement. At the end of thesis there are description of FW and SW which allow to acquire data to the computer.

Keywords

Acquire unit; pH; EC; liquid; UART; USB; C#

Obsah

ÚVOD.....	5
1 CO TO VLASTNĚ JE PH A EC?.....	5
1.1 CO JE TO PH.....	5
1.1.1 Molární hmotnost látky.....	6
1.1.2 Látkové množství.....	6
1.1.3 Koncentrace a aktivita elektrolytu.....	6
1.2 CO JE TO EC.....	6
1.2.1 Elektrolyt.....	6
1.2.2 Vztahy.....	6
1.2.3 Závislosti konduktivity a elektrolytu.....	7
1.2.4 Molární koncentrace neboli látkové množství.....	7
1.2.5 Molární konduktivita.....	7
1.2.6 Vliv teploty na elektrickou konduktivitu.....	7
2 MĚŘENÍ PH.....	8
2.1 MĚŘENÍ INDIKÁTOROVÝMI PAPIRKY.....	8
2.1.1 Měření pH pomocí elektrod.....	9
3 MĚŘENÍ EC.....	10
3.1 KONTAKTNÍ METODA.....	10
3.1.1 Dvou-elektrodová sonda.....	10
3.1.2 Čtyř-elektrodová sonda.....	11
3.2 NEKONTAKTNÍ METODA.....	11
3.2.1 Transformátorové a indukční sondy.....	11
2 ANALÝZA DOSTUPNÝCH SOND A METRŮ.....	11
2.1 PH METRY A SONDY.....	11
2.1.1 Komerční pH metry.....	11
2.1.2 Komerční pH sondy.....	12
2.1.3 Výběr sondy.....	12
2.2 EC METRY A SONDY.....	13
2.2.1 Komerční EC metry.....	13
2.2.2 Komerční EC sondy.....	13
2.2.3 Kombinované metry.....	14
3 OBLASTI POTŘEBY MĚŘENÍ PH A EC.....	15
3.1 CHEMICKÉ LABORATOŘE.....	15
3.1.1 Měření pH.....	15
3.1.2 Měření EC.....	15
3.2 VÝROBCI NÁPOJŮ A POTRAVIN.....	15
3.2.1 Měření EC.....	15
3.2.2 Měření pH.....	15
3.3 VÝROBCI KOSMETIKY.....	16
3.3.1 Měření pH u domácích produktů.....	16
3.4 PRŮMYSL OVÉ PĚSTOVÁNÍ ROSTLIN.....	16
3.4.1 Hlavní funkce EC metru.....	16
3.4.2 Hydroponie.....	16
3.5 BAZÉNY A AKVÁRIA.....	16
4 MIKROPROCESOR.....	17
4.1 VOLBA VHDNĚHO PROCESORU.....	17
4.2 PROCESOR ATMEGA8.....	18
4.3 VÝVOJOVÁ DESKA SPOLEČNOSTI ELEDUS S.R.O.....	18
4.4 ZAPOJENÍ PROCESORU PRO KONKRÉTNÍ APLIKACI.....	19

4.4.1	<i>Napájecí obvody</i>	19
4.4.2	<i>Krystal</i>	19
4.4.3	<i>AD převodník</i>	20
4.4.4	<i>UART</i>	20
4.5	FIRMWARE	21
4.5.1	<i>Vývojové prostředí</i>	21
4.5.2	<i>Obsluha UART</i>	21
4.5.3	<i>Obsluha AD převodníku</i>	23
5	SOFTWARE PRO AKVIZICI DAT	25
5.1	ZÁKLADNÍ POŽADAVKY NA APLIKACI	25
5.2	GRAFICKÉ ROZHŘANÍ APLIKACE	26
5.3	PRÁCE S PORTY	27
5.4	ZAHÁJENÍ A UKONČENÍ MĚŘENÍ	30
5.5	EXPORT DAT	30
5.6	VYKRESLOVÁNÍ GRAFU	31
5.7	ULOŽENÍ GRAFU DO OBRÁZKU	33
6	REALIZACE EC PŘEVODNÍKU	34
6.1	TEORETICKÁ ČÁST	34
6.1.1	<i>Napájecí obvod</i>	34
6.1.2	<i>Oscilátor</i>	35
6.1.3	<i>Zpracování signálu</i>	36
6.1.4	<i>Usměrnění signálu</i>	37
6.1.5	<i>Ochrana ADC</i>	37
6.1.6	<i>Důvod použití TL084CD</i>	37
6.2	PRAKTICKÁ ČÁST	37
6.2.1	<i>Simulace</i>	37
6.2.2	<i>Návrh a vytvoření desky</i>	38
6.2.3	<i>Měření EC</i>	39
7	REALIZACE PH PŘEVODNÍKU	40
7.1	TEORETICKÁ ČÁST	40
7.1.1	<i>Napájecí obvod</i>	40
7.1.2	<i>Operační zesilovač</i>	40
7.1.3	<i>Zesílení výstupního signálu sondy</i>	41
7.1.4	<i>Referenční napětí pro diferenciální zesilovač</i>	41
7.1.5	<i>Diferenciální zesilovač</i>	41
7.1.6	<i>Výsledný signál</i>	41
7.2	PRAKTICKÁ ČÁST	41
7.2.1	<i>Simulace</i>	41
7.2.2	<i>Návrh a vytvoření desky</i>	42
7.2.3	<i>Měření pH</i>	43
	ZÁVĚR	44
	LITERATURA	45
	SEZNAM OBRÁZKŮ	47
	SEZNAM VZORCŮ	48
	SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK	49
	SEZNAM PŘÍLOH	50

Úvod

První zmínka o pH sahá do 20 století, zmínka o EC až do 18 století. Měření pH se brzy začalo používat v různých oborech jako lékařství, potravinářství či k laboratornímu výzkumu. Při výzkumu a následném měření EC se přišlo i na princip akumulátor a na vztah elektrického proudu a vodivosti.

Obě tyto veličiny zasahují do našich životů a my o tom ani nevíme, jejich hodnoty se měří ve vodě, kterou pijeme, v potravinách ale i v zemině. Proto mě tento projekt zaujal natolik, že jsem se pustil do jeho zhotovení, chtěl jsem si dokázat změřit vše, co potřebuji, abych mohl například upravit vysazované rostliny na zahradě či abych věděl co je pro mě zdravé a co nezdravé.

Ze začátku práce si uvedeme, co jsou tyto veličiny, prostudujeme si jejich fyzikální principy, a jak se měří. Pozastavíme se i nad konstrukcemi měřících sond, jejich dostatky a nedostatky. Vysvětlíme si v jakých oblastech je vhodné tyto veličiny měřit či pozorovat a co nám usnadní.

Dále se budu zabývat komunikačním softwarem, popíšu jeho funkčnost a samostatné funkce a metody komunikace. Konstrukce pH a EC metru zasahuje do více oborů a proto vysvětlím nejen software tedy konečný program pro počítač ale i firmware pro mikroprocesor.

Ke konci práce znázorním, popíšu a vysvětlím realizaci obou metrů, jejich elektronické vlastnosti, samotné funkční bloky obsažené v obvodech a uvedu i použité komponenty. Cílem konstrukce je vytvořit takový přístroj, který dokáže měřit pH a EC v roztocích. Je zaměřený hlavně ke kontrole živící vody pro rostliny.

Po prostudování práce by měl být čtenář seznámen se základními principy a vztahy pH a elektrické vodivosti, s měřením těchto veličin, s novými oblastmi, které ani nemusel znát, kde se tyto veličiny měří a s návrhem a konstrukcí pH a EC metru jako pro stránce softwarové tak po stránce hardwarové.

1 Co to vlastně je pH a EC?

1.1 Co je to pH

Pro správné pochopení funkce pH sond, je nutné vědět, co to pH je a jaké hodnoty co znamenají. Samotné měření pH se pak uskutečňuje v jednom ze základních materiálů a to v roztocích.

„Roztokem se rozumí homogenní soustava, složená ze dvou nebo více druhů molekul, jejichž zastoupení lze alespoň v určitém rozmezí spojitě měnit.“(1)

1.1.1 Molární hmotnost látky

„Molární hmotnost elektrolytu je součet atomových hmotností všech obsažených atomů.“(2)

1.1.2 Látkové množství

Složení roztoků se vyjadřuje koncentrací rozpuštěných látek ve fyzikální jednotce, v molech na 1 litr roztoku. Mol je jednotka látkového množství. Kromě koncentrace roztoku můžeme definovat i další veličinu a to aktivitu roztoku, která se vyjadřuje stejně jako koncentrace v molech na litr.

Látkové množství se značí malým písmenem „ c “, k výpočtu je nutné znát váhu rozpuštěné látky (m_l) v gramech [g], molární hmotnost rozpuštěné látky ($M(l)$) v gramech na mol [g/mol] a nakonec objem (V) v litrech [l].

$$c = \frac{m_l}{M(l) * V_{H_2O}} \quad (1)$$

1.1.3 Koncentrace a aktivita elektrolytu

Koncentrace a aktivita roztoku (elektrolytu) slouží ke zjištění, zda je roztok kyselý či zásaditý. V praxi se obvykle na místo stanovení aktivity roztoku používá tzv. vodíkový exponent, který lze vyjádřit jako záporný dekadický logaritmus aktivity vodíkových iontů neboli,

$$„pH = -\log a_{H^+} \text{ ale i } pH = -\log c_{H^+}“.$$

Souvislost mezi koncentrací a aktivitou roztoku je následující, kde γ_x je tzv. aktivní koeficient složky X, tento koeficient nabývá hodnoty 0 či 1“.

$$a_x = \gamma_x * c_x \quad (2)$$

Existuje i tzv. elektrolyt, což je vlastně vodič, jehož proud je způsoben transportem iontů.

Hladina pH může nabývat hodnoty 0-14. Pokud je naměřena hodnota 0-7 pH, hovoří se o kyselinách. O zásady se jedná, pokud je hodnota pH 7-14. (1)

1.2 Co je to EC

EC neboli konduktivita. Tato veličina označuje elektrickou vodivost v roztocích. Konduktivita se měří v S/cm(Siemens na cm) a značí se řeckým písmenkem kappa (κ).

1.2.1 Elektrolyt

Pojem elektrolyt je využit i u měření konduktivity. Pokud se napájené elektrody vloží do elektrolytu, vznikne elektrické pole, ve kterém se začnou nabitě částice pohybovat k opačně nabitým elektrodám, čímž začne obvodem protékat proud. Tento vzniklý proud je přímo úměrný počtu nabitých částic v elektrolytu.

1.2.2 Vztahy

Podílem protékajícího proudu a napětí na elektrodách je možné získat elektrickou vodivost (G). Problém nastává, pokud chceme získat konduktivitu, jelikož je nutné zohlednit i

geometrické vlastnosti sondy. K výpočtu konduktivity tedy slouží tento vzorec „ $\kappa = G \cdot k$ “ kde „ k “ je konstanta sondy vypočítaná jako podíl vzdálenosti a plochy elektrod neboli „ $k = l/A$ “ [cm^{-1}]. Pokud mají být výše uvedené vztahy správné, musí být pole mezi elektrodami homogenní, toho se dosáhne například tím, že bude hladina stejně vysoká jako elektrody.

Bohužel v praxi tomu tak většinou není a proto se vztah pro výpočet konduktivity poněkud zkomplikuje. Pokud elektrické pole není homogenní, tak se konstanta obvykle určuje pokusem, který se skládá z připraveného roztoku (kalibračního), který má definovanou konduktivitu, a sondou u které není konstanta známá, posléze se změří vodivost tohoto roztoku touto sondou.

Konstanta sondy se získá podílem definované konduktivity a změřené vodivosti neboli.

$$k = \kappa / G \quad (3)$$

1.2.3 Závislosti konduktivity a elektrolytu

Konduktivita je závislá především na koncentraci a pohybu iontů, je však závislá i na teplotě elektrolytu.

1.2.4 Molární koncentrace neboli látkové množství

I zde se při měření konduktivity využije látkové množství. Tato fyzikální veličina nebude dále znovu rozepisována, jelikož se jí věnuje předchozí kapitola [1.1 Co je to pH + 1.1.1-1.1.2](#)

1.2.5 Molární konduktivita

Pro výpočet molární konduktivity bude zavedena iontová pohyblivost (λ_i). Pohyblivost iontů se mění spolu s teplotou elektrolytu. Některé změny pohyblivosti z důvodu rostoucí teploty je lepší nalézt v tabulkách. Molární konduktivita (Λ_0) se získá součtem pohyblivostí kladných a záporných iontů.

$$\Lambda_0 = \lambda_i^+ + \lambda_i^- \quad (4)$$

1.2.6 Vliv teploty na elektrickou konduktivitu

Buď lze konduktivitu vypočítat způsobem, kdy se pohyblivost iontů v dané teplotě vyhledá v tabulkách, anebo lze aplikovat tento vztah (obr. 1), kde Λ_t je molární konduktivita pro žádanou teplotu, β je koeficient teplotní závislosti, Λ_0 je molární konduktivita v nulové teplotě, t a t_0 jsou teploty, nulová a požadovaná.

$$\Lambda_t = \Lambda_0 \cdot (1 + \beta \cdot (t - t_0)) \quad [S \cdot \text{cm}^2 \cdot \text{mol}^{-1}]$$

Obrázek 1 - Vzorec pro výpočet EC při různých teplotách(1)

Pro výpočet konduktivity tímto způsobem je ještě nutné získat vztah pro závislost mezi pohyblivostí a molární koncentrací.

$$\kappa = \Lambda_t \cdot c \quad (5)$$

2 Měření pH

2.1 Měření indikátorovými papírky

Základní měřicí technika je založena na tzv. indikátorových pH papírcích (Obrázek 2), které se po vložení do roztoku a následném vymoření, zabarví určitou barvou. K indikátorovým pH papírkům bývá přiložen referenční seznam hodnot pH a k nim přiřazené barvy. Tato měřicí metoda je spíše odhadová a rozhodně není určena pro měření přesných hodnot.



Obrázek 2 - pH indikátory(21)

2.1.1 Měření pH pomocí elektrod

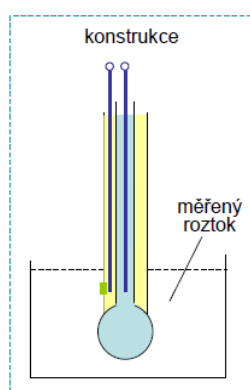
Měřicí technika, která je určena pro přesné měření, a zároveň metoda použitá v daném případě je založena na měření rozdílu galvanického napětí na dvou elektrodách. Jedna z těchto elektrod je referenční a druhá je vystavena měřenému roztoku.

Sonda je tvořena referenční elektrodou, která je ponořena většinou do roztoku stříbra neboli chloridu stříbrného, a elektrodou skleněnou, ta je vystavována měřenému roztoku. Měřicí elektroda musí být skleněná, aby měla co nejmenší citlivost na H^+ v roztoku.

„Z termodynamických úvah plyne, že pokud je postupně změřeno elektromotorické napětí článku tvořeného referenční elektrodou a pH-elektrodou v roztoku A (U_A) a v roztoku B (U_B), potom rozdíl pH obou roztoků ($pH_A - pH_B$) je přímo úměrný rozdílu

naměřených napětí $U_A - U_B$.“(1)

Přesnost měření se odráží od kvality elektrod a elektroniky v použitém přístroji.



Obrázek 3 - Skleněná elektroda(1)

Před samotným měřením je nutné sondu zkalibrovat a to pomocí tzv. pufrů (kalibrační roztoky) většinou s hodnotami 4, 7 a 10 pH. Pro danou úlohu stačily pufrы s hodnotou 4 a 7 pH.

2.1.1.1 Postup kalibrace

Kalibrace probíhala tak, že se pufrы nechaly zahřát na pokojovou teplotu, což je velmi důležité z důvodu zkreslení měřené hodnoty. Teprve poté bylo možné sondu připojit k elektronice a sundat z ní ochranou baňku s KCl (Chlorid Draselný). Kalibrace musí probíhat tak, aby se pufrы nesmíchaly a proto se musí sonda před dalším ponořením vyčistit a vysušit, což lze provádět obyčejnou vodou a papírovými ubrousky.

Postup kalibrace na samotném elektrickém zařízení (viz 7.2.3).

2.1.1.2 Údržba sondy

Sonda respektive její elektrody nesmí zůstat suché. Referenční elektroda je uzavřená v sondě, kde se nachází roztok s referenčním pH, avšak při delší neaktivitě může elektroda vystavovaná měřenému roztoku, zaschnout a poškodit se. Proto mají některé sondy vzduchotěsný uzávěr s KCl (Chlorid Draselný). Sonda nesmí být vystavena vysokým teplotám. Pokud je sonda správně udržována, vydrží i řadu let. Postupem času jak elektroda stárne, je nutné ji průběžně kalibrovat. Problematika kalibrace zatím není v této práci plně dořešena, avšak vzhledem k její důležitosti bude předmětem dalšího výzkumu.

3 Měření EC

Konduktivitu je možné měřit kontaktním či nekontaktním způsobem. Kontaktní způsob je založen na ponoření dvou či čtyř elektrodových sond do roztoku. Pokud je roztok agresivní, musí být elektrody vyrobeny z chemicky odolných materiálů.

3.1 Kontaktní metoda

Tuto metodu lze použít pro měření od 10^{-8} S/cm do 1 S/cm. Při měření se projevuje nežádoucí efekt polarizace elektrod, což se projevuje jako kondenzátor s měřeným roztokem, tento efekt se dá částečně vyrušit měřením se střídavým proudem. Kontaktní metoda byla použita pro měření v případě této práce. Pro přesnější hodnotu EC je potřeba se sondou párkrát promíchat roztok, aby došlo k redukci počtu bublinek v roztoku.

3.1.1 Dvou-elektrodová sonda

Jedná se o nejjednodušší sondu, která lze využít pro měření konduktivity. Sonda je napájena střídavým elektrickým proudem a měří se na ní elektrické napětí.

Problémem u dvou-elektrodových sond je ten, že se na elektrodách tvoří parazitní kapacita, což lze částečně eliminovat snížením protékajícího proudu a také zvýšením frekvence.

Kohlrausch při svých experimentech zjistil, že pro eliminaci polarizačního efektu dvou-elektrodové sondy je ideální „pokovení elektrod vrstvou černé platiny“(2)

Řešení dvou-elektrodové sondy používám ve své práci já.

3.1.2 Čtyř-elektrodová sonda

Tato sonda obsahuje dvě proudové elektrody (stejně jako u dvou-elektrodové sondy), ke kterým se přidají další dvě napěťové. Na proudových elektrodách se sice bude vytvářet nežádoucí kapacita, ale nebude se na nich měřit napětí. Napětí se měří zvlášť na napěťových elektrodách. Konduktivita je v tomto případě vypočítána stejně jako u dvou-elektrodové sondy.

3.2 Nekontaktní metoda

Existuje i nekontaktní metoda kdy sonda nepřijde s roztokem do kontaktu. Díky tomu odpadá potřeba pokrývat elektrody chemicky odolnými materiály.

3.2.1 Transformátorové a indukční sondy

Je-li přivedeno střídavé napětí na primární cívku transformátoru, vybudí se tím proud v elektrolytu, který je přímo úměrný budícímu napětí i vodivosti elektrolytu, která je určena jak vlastnostmi elektrolytu, tak i geometrií závitu na krátko na druhém transformátoru, v jehož sekundární cívice se indukuje napětí přímo úměrné vodivosti elektrolytu.

4 Analýza dostupných sond a metrů

4.1 pH metry a sondy

4.1.1 Komerční pH metry

Komerčních měřičů je velké množství, proto bude uveden vždy jen jeden na každou skupinu. Rozdělit pH metry je možné podle přesnosti, rozsahu, velikosti či ceny. Některé mají i tepelnou kompenzaci s nastavitelným faktorem. U některých se provádí kalibrace manuálně a u jiných zase automaticky. Jeden faktor mají však společný, a tím je stárnutí elektrody. Z tohoto důvodu je vhodnější, poohlížet se spíše po pH metru s vyměnitelnou sondou. Levnější pH metry v cenové kategorii kolem 500 Kč zpravidla nemívají vyměnitelnou sondu ani teplotní kompenzaci a při správné údržbě vydrží něco kolem jednoho roku, poté je nutné koupit celý měřič znovu. V případě dražších pH metrů je funkce a výdrž sondy stejná jako u těch levnějších s tím rozdílem, že se vyměňuje pouze sonda a že mívají větší přesnost měření.



Obrázek 4 - dražší metr (4)



Obrázek 5 - levný metr (3)

4.1.2 Komerční pH sondy

Dostupné náhradní sondy na trhu se liší především podle velikosti vnitřního odporu sondy a typu konektoru. Odolnější sondy jsou schopny při správném používání a údržbě vydržet až 2 roky.

4.1.3 Výběr sondy

Pro účely této práce byla pohledávána sonda s přesností 0,2 – 0,5 pH, teplotním rozsahem až 40°C a cenovým limitem do 1000kč. Zejména kvůli cenovému limitu bylo nutné hledat vhodnou sondu na zahraničním trhu. Nakonec byla pořízena sonda z Číny, u které sice nebyl přesný návod, avšak po navržení elektroniky se volba, z hlediska počátečních požadavků, ukázala jako vhodná. Risk z objednání cenově přístupné sondy ze zahraničního trhu vyvážil přínos v podobě získání praktických zkušeností, vytvořením vlastního výrobku a tím i dobrou znalostí nedostatků, které bude případně nutné vylimovat v dalších fázích vývoje, což by u zakoupeného měřiče nebylo možné.

Sonda má konektor BNC a její výstup závisí na hodnotě pH v měřeném roztoku, spolu s hodnotou pH se mění napětí vytvořené sondou. (5)

Sonda by měla být lineární až do chvíle kdy je v roztoku nízká koncentrace vodíkových iontů neboli když je hodnota pH velmi vysoká, dochází k tzv. alkali error range kdy potenciál pH sondy začne být nelineární.



Obrázek 6 - sonda pH(5)

4.2 EC metry a sondy

4.2.1 Komerční EC metry

EC metry tak jako pH metry je možné rozdělit podle ceny na levné, drahé a podle kvality měření na přesné a nepřesné. Cena EC metrů na českém trhu začíná na hranici 1000 Kč a může se vyšplhat až na cenu kolem 10 000 Kč. Pro účely dané práce by bylo dostatečné zakoupit EC metr v nižší cenové hladině. Avšak vzhledem ke zkušenostem získaných s prací na pH sondě a dalších výhod vyplývajících pro další vývoj a také výhody v podobě nových zkušeností bylo dospěno k rozhodnutí vytvořit i celý EC metr.

4.2.2 Komerční EC sondy

Vzhledem k dobré předchozí zkušenosti s nakupováním na zahraničních obchodech na Aliexpress, byla i EC sonda pohledávána na zahraničním trhu. Jednou z nevýhod byly neodpovídající popisy či častá absence zboží na skladě, což však substituuje velký výběr. Z hlediska zamezení teplotního zkreslení při měření EC, se kterým je nutné počítat, byla zakoupena sonda s termistorem.



Obrázek 7 - EC sonda(6)

4.2.3 Kombinované metry

Na trhu se objevují také metry kombinované, to znamená metry měřící pH i EC zároveň.

Pohybují se v cenové relaci asi 5500 Kč a výše, většinou se jedná o profesionálnější zařízení.

5 Oblasti potřeby měření pH a EC

5.1 Chemické laboratoře

5.1.1 Měření pH

V laboratorních podmínkách se volí jednoduchá metoda měření pH a to pomocí lakmusových papírků, které byly zmíněny ve druhé kapitole o měření pH. Další, méně používanou, ale přesto dosti zajímavou metodou je měření pH pomocí antokyanů.

Antokyaniny jsou přírodní látky obsaženy v rostlinách (např. červené zelí). Princip je velice jednoduchý. Z rostliny se udělá výluh (metoda funguje pouze u rostlin s fialovou barvou), do něhož se po kapkách přidávají látky, které podle kyselosti zbarví roztok s výluhem. Tato metoda nepatří mezi nejpřesnější a řadí se spíše mezi biologické metody.

5.1.2 Měření EC

Jako příklad měření EC v laboratořích nám může posloužit, měření solnatosti půdy. Postup měření je takový, že se buď vezme vzorek půdy, který se pak testuje v chemické laboratoři, kde jsou k dispozici přesné a velmi drahé EC metry, kterými je možné změřit přesnou hodnotu v S/cm. Nebo postačí méně přesný postup s pomocí levného EC metru, u kterého se odebere vzorek půdy a následně se smíchá s vodou. U takto vzniklého roztoku se pak jen změní hodnota EC.

5.2 Výrobci nápojů a potravin

5.2.1 Měření EC

Výrobci nápojů potřebují kontrolovat pH i EC hodnoty u vstupního roztoku (většinou vody) a u výsledného výrobku. Voda do 800 $\mu\text{S}/\text{cm}$ je pro člověka dobře požitelná, kolem této hodnoty se pohybuje obyčejná voda z kohoutku. Voda v moři má hodnotu asi 50 000 $\mu\text{S}/\text{cm}$ a může být použita ke splachování na toaletách, avšak pouze pokud může být koroze v cisternách či v trubkách kontrolována, protože voda s vysokým obsahem soli působí silně korozivním účinkem. Pokud je to životně nutné, člověk dokáže vypít vodu či jiný roztok až do hodnoty 3000 $\mu\text{S}/\text{cm}$. Voda s vyšším obsahem soli neboli s hodnotou nad 3000 $\mu\text{S}/\text{cm}$ není pro člověka vhodná, některá zvířata však dokáží pozřít vodu do hodnoty až 17 000 $\mu\text{S}/\text{cm}$. Je zde ovšem riziko že voda či jiný roztok s tak vysokou hodnotou EC může obsahovat nepřijatelnou koncentraci konkrétních iontů.

Hlavním důvodem měření EC v potravinářství je tedy stanovení obsahu soli v jednotlivých potravinách z hlediska chuti, požitelnosti a splnění norem.

5.2.2 Měření pH

Potraviny, které konzumujeme, mají určitou hodnotu pH. Velkovýrobci potravin v této době vyrábí spíše kyselé jídlo, což není zdravé pro lidské tělo. Lidské tělo by mělo mít hodnoty kolem 6 – 7,5 pH. Kyselé jídlo může způsobit zdravotní problémy, jako třeba problémy se srdcem či slabé kosti, v případě, že dojde k okyselení těla. Člověk by měl konzumovat spíše potraviny, které mají hodnoty do 7 – 8 pH, jedná se hlavně o zeleninu, ovoce. I když například nakládané ovoce či oříšky se řadí do velmi kyselých potravin. Kyselost v těle si člověk zvyšuje i konzumací alkoholu, kávy nebo i černého čaje. Mastné potraviny se také řadí spíše do kyselých potravin.

Pro úplnost je však nutné znát, jestli je potravin kyselinotvorná či zásadotvorná. Například citrón, který patří do kyselých potravin, tvoří v těle spíše zásadu, takže je zásadotvorný. Důvod proč jsou některé potraviny kyselé, ale v těle se chovají zásadotvorně, je

jednoduchý. Při růstu jsou plodiny ovlivňovány okolím a jsou-li vystavovány kyselému prostředí (kyselá půda či voda), odrazí se to na jejich výsledné kyselosti či zásaditosti, avšak až obsah určitých látek určí, bude-li se jednat o potraviny zásadotvorné, nebo kyselinotvorné. Pokud potravina obsahuje více vápníku, sodíku či draslíku je zásadotvorná, pokud však obsahuje více síry, chloridu či fosfátu je kyselinotvorná.(8)

5.3 Výrobci kosmetiky

Měřit pH je zapotřebí i u výroby kosmetiky. Lidská pokožka je na kyselost velmi citlivá a proto je nutné hlídat hodnotu pH různých kosmetických výrobků (např. různá mýdla, šampony, krémy, atd.).

5.3.1 Měření pH u domácích produktů

Pokud si chceme vyrobit vlastní mýdlo či jinou kosmetiku, je potřeba vědět, jakou má výrobek pH hodnotu a podle ní se řídit při výrobě. Rozsah pH u takového mýdla by měl být něco mezi 7 až 9 pH. Na škodu rozhodně není ani měření již zakoupených výrobků, zejména u jedinců s precitlivělou pokožkou či kožními problémy. Dosti často se stává, že kosmetika zakoupená v obchodech má na obalech uvedenou odlišnou hodnotu pH, než jak je tomu ve skutečnosti.

5.4 Průmyslové pěstování rostlin

Při průmyslovém pěstování rostlin je nutné sledovat parametry vody, kterou se rostliny zalévají z důvodu vytvoření vhodného prostředí, čímž se zajistí správný příjem živin a odstraní se tak případné nežádoucí problémy spojené s nesprávným prostředím (odumírání, předčasné stárnutí, některé nemoci, špatné výnosy, atd.).

5.4.1 Hlavní funkce EC metru

EC metr měří koncentraci solí v roztoku, díky které lze spolehlivě odhadnout množství živin obsažených v roztoku. Při příliš vysoké hladině EC se může stát, že rostlina odumře z důvodu přebytku živin a naopak při nízké hodnotě EC může uhynout kvůli nízké koncentraci živin v roztoku.

5.4.2 Hydroponie

Hydroponie se zabývá pěstováním rostlin mimo půdu (tzn. v živném roztoku) a v průmyslovém pěstování zeleniny se jedná o převažující metodu (pokud nepočítáme sezonní pěstování na poli). EC metr v tomto případě je nutný, jak již bylo uvedeno výše, ke kontrole míry živin v roztoku. Měření pH pak slouží pro ověření, zda je voda, ve které se rostlina nachází zásaditá či kyselá, což má vliv na schopnost rostliny přijímat živiny z roztoku.

Pěstování v tomto systému má velkou výhodu v tom že, lze pěstovat i ve městech v uzavřených halách a průmyslových sklenících. Z hlediska neustále rostoucího zamoření a znečištění přirozené půdy a z hlediska vyšší efektivity se předpokládá velký nárůst tohoto způsobu pěstování (zejména oblastech po ekologických katastrofách, zdevastované oblasti špatným způsobem zemědělského obhospodařování, pouště, velké aglomerace bez volné půdy, atd.).

5.5 Bazény a akvária

V bazénech je nutné měřit pH z důvodu citlivosti lidské pokožky. Hodnota pH v bazénové vodě by se měla pohybovat v rozmezí 7 – 7,4 pH, což upravují jednotlivé vyhlášky.

Pokud má voda extrémně nízkou hodnotu EC (měkká voda- horské prameny, dešťová voda, atd.) způsobuje to podráždění pokožky a její následné vysychání až popraskání.

V akvaristice se pH metr používá z důvodu požadavků ze strany chovaných rybiček a pěstování podvodních rybiček.

6 Mikroprocesor

6.1 Volba vhodného procesoru

Mikroprocesor je v dnešní moderní elektronice základem každého zařízení. Umožňuje velmi flexibilně pracovat se vstupy a výstupy, provádět matematické operace, komunikovat po různých typech kanálů, apod. Protože je v dnešní době nepřehledné množství

mikroprocesorů, je potřeba si prvně stanovit požadavky dané aplikace a dle nich vyselektovat vhodný mikroprocesor.

V dané aplikaci je potřeba provést následující:

- Načítat analogové hodnoty (minimálně ze 2 kanálů)
- Přeposlat hodnoty do počítače

Požadavky dané aplikace splní většina běžných procesorů. Autor této práce má nejvíce zkušeností s procesory od značky Atmel (konkrétně rodiny AVR), proto jako první procesor přišel v úvahu ATmega8. V rámci společnosti ELEDUS byla k zapůjčení poskytnuta vývojová deska s tímto procesorem, díky čemuž testování zabralo podstatně kratší dobu.

ATmega8 poskytuje 8KB flash paměti pro uložení operačního kódu. Další varianty ATmega se většinou liší primárně prostorem v paměti a větším počtem periférií. V případě potřeby je tedy možné přenést základ kódu i na vyšší řady procesorů, pokud by byly dosaženy limity dané verze procesoru ATmega8. V dané aplikaci se však tento problém neočekává.

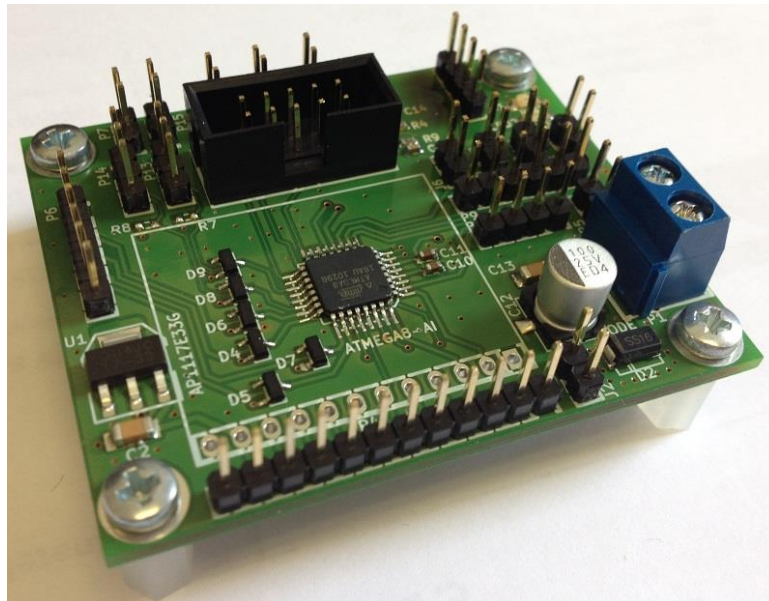
6.2 Procesor ATmega8

Jedná se o jeden z nejzákladnějších procesorů z rodiny AVR. Je velmi dobře dostupný cenou i dokumentací. Detailní popis zde však není potřeba uvádět, protože již o dané tématice bylo napsáno mnoho článků.(9)

Tento typ procesoru disponuje (mimo jiné) 8 kanálovým analogově-digitálním (dále AD) převodníkem a integrovaným driverem pro UART (sériová linka), kterou je možné velmi jednoduše spojit s počítačem pomocí integrovaných USBtoUART převodníků, které je dnes možné velmi levně pořídit.

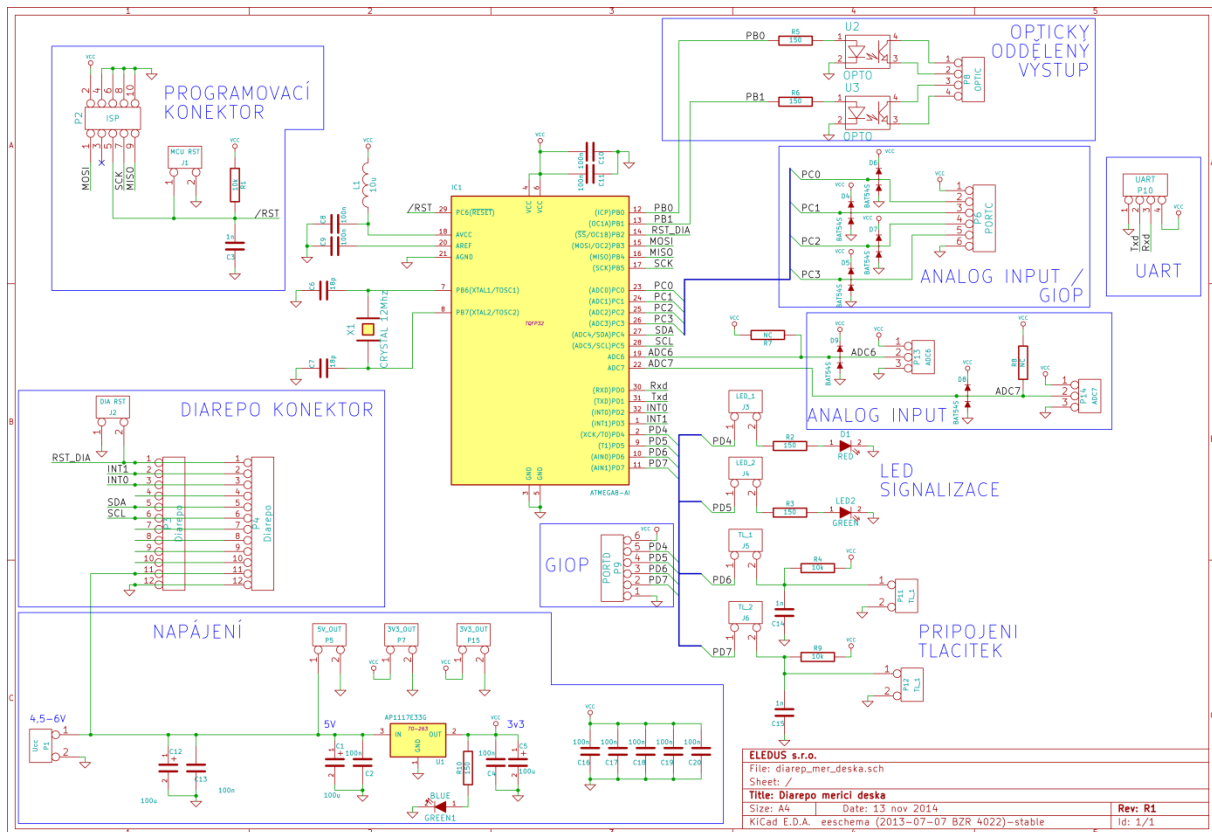
6.3 Vývojová deska společnosti ELEDUS s.r.o.

Vývojová deska s procesorem vznikla v rámci konkrétního produktu, který již nyní není na trhu. Nicméně univerzálnost dané desky umožňuje široké uplatnění ve stádiu vývoje u řady projektů. Tento projekt nebyl výjimkou, proto byla daná deska použita ve vývojových fázích. Vývojové desky poskytují možnost vše důkladně otestovat před samotnou realizací vlastního návrhu kompletního zařízení.



Obrázek 8 – Vývojová deska s ATmega8

Na obrázku je vidět vývojová deska, která obsahuje stabilizátor pro úpravu napájení na 3,3V, ESD ochrany na vstupech AD převodníku, programovací konektor ISP10, dva výstupy oddělené optočleny, integrované LED a ostatní vstupně výstupní piny. Schéma desky je na následujícím obrázku:



Obrázek 9 – schéma vývojové desky s ATmegou8

6.4 Zapojení procesoru pro konkrétní aplikaci

6.4.1 Napájecí obvody

V dané aplikaci je lepší použít pro napájení procesoru 5V. Vývojová deska standardně obsahuje stabilizátor pro převod z 5V na 3,3V. Tento stabilizátor byl přemostěn, aby se na procesoru dosáhlo přímo 5V ze vstupu. Filtrační kondenzátory však byly ponechány. Při vývoji FW a SW byla deska v mnoha případech napájena z USB přes převodník USBtoUART. Ve finální aplikaci i se zesilovači je vše napájeno centrálním zdrojem.

6.4.2 Krystal

Deska má vlastní krystal, který taktuje procesor na 12MHz. Pokud by daná frekvence nevyhovovala, je možné použít ještě interní oscilátor procesoru, **který však umožňuje maximální frekvenci.**

Procesor ATMega8 je možné taktovat maximálně na frekvenci 16MHz. U finální desky se doporučuje použít co největší možnou frekvenci, aby bylo možné dosáhnout maximální možné snímací frekvence na AD převodníku.

6.4.3 AD převodník

AD převodník je možné referovat vůči integrované napěťové referenci 2,56V nebo vůči externí referenci až 5V. Na vývojové desce je použita externí reference na hlavní napájecí větvi. Aby se omezil přenos rušení, je reference k procesoru přivedena přes LC filtr z tlumivky L1 a kondenzátoru C8. Protože je na vývojové desce umístěn procesor v pouzdře TQFP, je možné využít až 8 10 bitových kanálů na AD převodníku. Tyto piny jsou multiplexovány integrovaným multiplexerem, takže není možné snímat ze všech pinů hodnoty najednou, ale je potřeba je vždy přepnout. Pro danou aplikaci byly použity piny ADC0 a ADC1 (PC0 a PC1).

Na vývojové desce jsou integrovány ochrany proti ESD, tudíž by nemělo dojít k poškození vstupů AD převodníku ani v případě, že dojde k vybití statického výboje do pinů na desce.

6.4.4 UART

Pro komunikaci je nejvhodnější použít sériovou linku. Jak již bylo zmíněno výše, pro daný typ komunikační linky je možné velmi levně pořídit převodníky na USB sběrnici, což výrazně zjednoduší spojení s počítačem, který by měl sloužit pro ukládání dat.

Konkrétně použitý převodník je založený na čipu PL2303HX:



Obrázek 10 - USB to UART rozhraní

Na finálním zařízení je možné převodník integrovat přímo na desku například v podobě čipu FT232 od společnosti FTDI. Pro účely vývoje je však mnohem flexibilnější tato varianta, protože tyto čipy jsou poměrně drahé (kolem 130Kč/ks), takže umístit je na nějaké prototypové desky se může dost prodražit, v případě, kdy dojde k nějakým problémům.

Převodník má 4 piny s následujícími funkcemi:

- Červený: 5V z USB
- Černý: GND
- Bílý: RxD
- Zelený: TxD

Pro převodník existují drivery pro Windows i Linux. Na Windows se ovladače nainstalují samostatně a není potřeba je instalovat ručně na rozdíl od čipů FTDI, kde většinou Windows nenainstaluje ovladač správně.

7 Firmware

7.1.1 Vývojové prostředí

Jako vývojové prostředí jsem použil Atmel Studio 7. Jedná se o nejnovější verzi Atmel studia, které je založeno na enginu Microsoft Visual Studio, které je velmi dobře propracované a umožňuje velmi přehledně vyvíjet kód. Atmel studio má také velkou výhodu v podobě ASF Wizardu (10), který umožňuje velmi rychle přidat různé před-programované drivery a tím šetří čas při vývoji.

7.1.2 Obsluha UART

Sériová linka na procesoru ATMega8 se konfiguruje pomocí několika registrů. Přesný popis všech způsobů nastavení je uvedený v datasheetu procesoru (11). Zde je uvedeno konkrétní nastavení pro danou aplikaci. V aplikaci byly zvoleny následující nastavení sériové linky:

- Mód komunikace: Asynchronní
- Přenosová rychlost: 38400 baud
- Počet datových bitů: 8
- Stop bity: 1
- Parita: žádná

Tyto hlavní parametry je potřeba nastavit v procesoru. Pro volbu slouží následující registry:

- UBRR – nastavení přenosové rychlosti
- UCSRB – povolení příjmu a odesílání dat
- UCSRC – nastavení formátu dat

Pro nastavení rychlosti je potřeba spočítat hodnotu, která se nastaví do UBRR:

$$UBRR = \frac{FOCS}{16 * BAUD} - 1 \quad (6)$$

Kde FOCS je taktovací frekvence procesoru (v daném případě 12MHz) a BAUD je přenosová rychlost (v daném případě 38400).

Následující vzorec je implementován přímo v definici na začátku programu, aby nebylo nutné ručně počítat a zadávat hodnotu UBRR. Stačí jen v příslušném definici změnit hodnotu taktovací frekvence nebo přenosové rychlosti. Jedná se o následující část kódu:

```
#define FOCS 12000000
#define BAUD 38400
#define MYUBRR ((FOCS/16/BAUD)-1)

Celkovou inicializaci sériového portu zajišťuje následující funkce:
void USARTInit(uint16_t ubrr_value)
{
    //Set Baud rate
    UBRRH = (unsigned int)(ubrr_value>>8);
    UBRRL = (unsigned int)ubrr_value;

    //Set frame format
    UCSRC |= (1<<URSEL)|(1<<UCSZ0)|(1<<UCSZ1);

    //Enable The receiver and transmitter
    UCSRB |= (1<<RXEN)|(1<<TXEN);
}
```

Při inicializaci je ještě nutné hodnotu MYUBRR rozdělit na dvě části, protože se jedná o 16 bitové číslo. Následně je nastaven mód přenosu a nakonec se povolí příjem a odesílání.

Pro samotnou komunikaci pak v kódu byly definovány tyto funkce:

```
void USARTWriteChar(char data)
{
    //Wait until the transmitter is ready
    while(!(UCSRA & (1<<UDRE)))
    {
        //Do nothing
    }
    //Now write the data to USART buffer
    UDR=data;
}
```

Tato funkce odešle jeden znak na UART. Prvně čeká, dokud není připravené odesílání (což je důležité například pro případ, když by zrovna probíhal příjem nějaké zprávy).

S touto funkcí souvisí i funkce pro posílání řetězce dat:

```
void USARTSendString(char *s)
{
    while (*s) //continue while actual address contains data
    {
        USARTWriteChar(*s++);
        //send actual data at address and move pointer to next address
    }
}
```

Funkce dostane určité pole znaků, které pokud je správně inicializované, obsahuje na konci ukončovací null znak. Funkce vždy odešle znak z aktuální adresy a posune ukazatel na další buňku v paměti. Pokud dorazí na null znak, tak smyčka skončí a tím je zároveň dokončen i přenos celého řetězce.

Poslední je funkce pro čtení dat:

```
char USARTReadChar()
{
    //Wait until a data is available
    while(!(UCSRA & (1<<RXC)))
    {
        //Do nothing
    }
    //Now USART has got data from host
    //and is available is buffer
    return UDR;
}
```

Tato funkce má jako jediná návratovou hodnotu. Vrací znak, který přijde do bufferu. Tato funkce čeká, dokud na port nepřijdou nějaká data. Pokud by nikdy nepřišla data na port, tak funkce bude stále čekat a nikdy neskončí, což je potřeba mít na paměti! V našem případě je tento problém však užitečný, protože umožňuje velmi jednoduše taktovat akvizici dat ze softwaru, který ve vhodný čas pošle určitý znak do procesoru a tím si vyžádá přečtení dat z AD převodníku.

Po definici těchto funkcí je již práce s portem velice jednoduchá. Prvně je potřeba inicializovat port zavoláním:

```
USARTInit((uint16_t)MYUBRR); //UBRR sets value for
```

Kde hodnota MYUBRR je spočítána automaticky makrem, které jsem bylo vysvětleno výše.

Pro posláni jednoho znaku například odřádkování stačí zavolat:

```
USARTWriteChar('\n'); //send new line character to port
```

Pro posláni řetězce se používá funkce:

```
USARTSendString("01,"); //send name of value to port
```

V případě, že je potřeba získat určitý znak z portu, je nutné vyvolat funkci pro čtení, které ještě musí být řečeno, kam má uložit data:

```
char data;
//Read data
data = USARTReadChar();
```

Pro uložení dat je potřeba mít deklarovanou proměnou typu char, stejně jako vrací funkce USARTReadChar().

V případě, že by bylo potřeba načíst více znaků, je možné tyto znaky uložit do pole char. V našem případě to však není třeba, protože tato funkce slouží jen pro synchronizaci se softwarem.

7.1.3 Obsluha AD převodníku

AD převodník je potřeba nastavit podobně jako tomu je u UARTu. Vše se nastavuje pomocí konkrétních registrů. Všechny možnosti nastavení zde nebudou rozebírány, jelikož je možné si je přečíst v datasheetu k ATmega8 (11). U dané aplikace stačí na AD převodníku nastavit jen tyto parametry:

- Referenční napětí: Externí
- Dělič frekvence: 32
- Zdroj signálu: ADC0 nebo ADC1

Referenční napětí a zdroj signálu se nastavují v rámci jednoho byte. Jedná se o byte ADMUX. Jeho nastavení je pro externí referenci a signál z ADC0:

```
ADMUX=0x40;
```

Pro externí referenci a signál z ADC1:

```
ADMUX=0x41;
```

Tímto je možné i přepínat mezi kanály. Způsob jak přenastavovat kanály byl řešen přímo ve funkci s žádostí o navrácení hodnoty z převodníku (viz níže).

Nastavení děliče a povolení AD převodníku proběhne hned na začátku main funkce.

V programu na něj není samostatná funkce, protože se jedná jen o jeden řádek:

```
ADCSRA = (1<<ADEN) | (1<<ADPS2) | (1<<ADPS0);
```

ADEN povolí používání AD převodníku a ADPS2 a ADPS0 nastaví děličku frekvence na 32. Tabulka s dalšími nastaveními před děliček je v datasheetu k ATmega (11) na straně 201. Pro získání hodnoty z převodníku byla sestavena následující funkce:

```
unsigned int getADCValue(char channel)
{
    unsigned int adc_value = 0;
    ADMUX=0x40 | channel;

    ADCSRA |= (1<<ADSC); // Start conversion
    while (ADCSRA & (1<<ADSC)); // wait for conversion to complete
    adc_value = ADCW; //Store ADC value

    return adc_value; //return ADC value
}
```

Funkce vrací hodnotu typu int, která obsahuje 10 bitové číslo z AD převodníku. Funkci se předává jeden parametr, a to číslo AD kanálu, který pro danou vývojovou desku může být 0 až 7. Toto číslo se spojí s přednastavenou hodnotou ADMUX, která nastavuje externí referenci. Poté se spustí AD převod a počká se do nastavení bitu indikující dokončení. Poté se již vrátí získaná hodnota.

V programu se použije funkce následovně:

```
unsigned int adc_value; // Variable to hold ADC result
adc_value = getADCValue(0); //get value from pH probe
```

Prvně je potřeba vytvořit si proměnnou pro uložení hodnoty. Poté se jen zavolá funkce s číslem kanálu, ze kterého je nutné číst.

Pro odeslání hodnoty přes UART je ještě potřeba ji převést na pole znaků, k čemuž se dá použít funkce itoa(int in,char* out,int base).

```
itoa(adc_value,string,10); //convert adc value to string
USARTSendString("00,");
USARTSendString(string);
USARTSendString(";"); //send semicolon to port
```

Tato série funkcí převede hodnotu z převodníku například hodnotu 1024) a pošle ji na UART ve tvaru:

```
„00,1024;“
```

Příkaz je tedy složen z identifikace kanálu (zde 00), hodnoty a ukončovacího znaku. Tyto znaky umožňují jednoduše separovat číslo ze zprávy, jak je ukázáno níže u popisu SW.

Celý kód pro procesor je poměrně jednoduchý a proto nebylo složité jej zoptimalizovat. Kód je přiložen k práci. Pro správnou funkci je ještě potřeba přenastavit pojistku v procesoru, která určuje volbu oscilátoru. Toto je možné jednoduše provést například v programu Khazma Programmer (12), kde se načtou aktuální nastavení procesoru a v roletovém menu se vybere možnost „Ext. Crystal/Resonator High-freq“. Startovací čas je většinou lepší nastavit na nejdelší.

8 Software pro akvizici dat

Pro variabilnější práci s daty bylo vhodné vytvořit vlastní software, který dokáže zobrazit data v grafu. V běžných programech pro komunikaci je možné získat konkrétní hodnoty, ale překreslování do grafu (například přes Excel) je zbytečně zdlouhavé a neumožnilo by pohodlné dlouhodobé sledování hodnot.

Pro tvorbu softwaru byl zvolen jazyk C# a to ze dvou důvodů. Jednak se autor této práce s daným jazykem více seznámil při spolupráci se společností ELEDUS a také díky vývojovému prostředí Microsoft Visual Studio, na kterém je postavené i Atmel Studio 7, takže práce v něm je velmi obdobná.

Původně byla zvažována tvorba aplikace v klasickém frameworku Windows Forms, které však již nyní vycházejí z módy a mají menší podporu v komunitě. Proto konečné rozhodnutí padlo na vytvoření aplikace pod novější verzí frameworku formulářů a to Windows Presentation Foundation (dále WPF). WPF poskytují mnohem lepší práci se vzhledem aplikace a obecně práci s grafikou.

Sice pro danou aplikaci nebyl dáván důraz na vzhled, avšak díky tomu, že bude aplikace ve WPF, je zde do budoucna možnost rozvíjet práci i formou upgradu vzhledu aplikace, čímž se dosáhne zlepšení uživatelské přívětivosti celého systému.

8.1 Základní požadavky na aplikaci

Pro sestavení aplikace bylo potřeba v první fázi navrhnout hlavní funkce, které se od softwaru očekávají. Nutno podotknout, že software je značně rozsáhlý, takže vzhledem k termínu odevzdání práce není ještě v konečném rozsahu, nicméně již nyní plní většinu potřebných funkcí, díky kterým je možné sledovat požadované hodnoty a určitým způsobem s nimi pracovat. Práce bude navíc dále pokračovat v rámci maturitní práce, kde dojde

k dalšímu rozvoji softwaru a pokud se podaří z projektu udělat i produkt, tak je možné, že se práce rozvine dále i po dokončení maturitní práce.

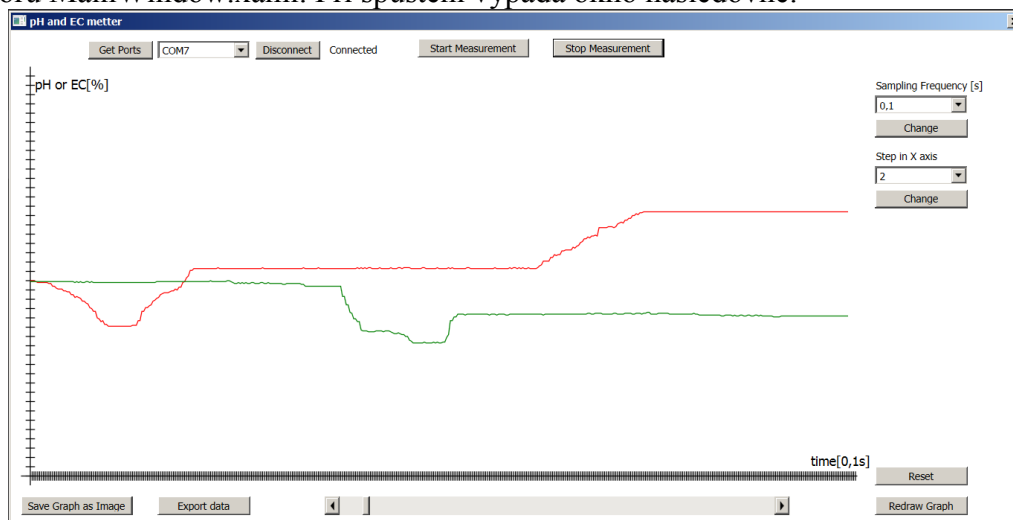
Nyní však zpět k návrhu hlavních funkcí. Mezi nejnужnější funkce stanovené pro zhotovení v rámci práce patří následující:

- Vytvoření hlavního okna
- Obsluha komunikačního portu a zpracování dat
- Funkce pro vykreslení grafu
 - o Vykreslení os grafu
 - o Vykreslení spojnic naměřených dat
- Ukládání grafu v podobě obrázku
- Změna smplovací frekvence
- Změna rozlišení dat v grafu

Výše uvedené funkcionality budou postupně vysvětleny v následující části práce. Celý kód je přílohou k této práci.

8.2 Grafické rozhraní aplikace

Okno je velmi jednoduché. Jeho kód zde přiložen není, ale je možné jej nalézt v souboru MainWindow.xaml. Při spuštění vypadá okno následovně:



Obrázek 11 - GUI rozhraní

V horní části okna jsou tlačítka pro získání dostupných portů, list se seznamem portů a tlačítka pro připojení či odpojení portu. Jeho nápis se mění podle toho, jestli je zrovna port připojený nebo není. Vedle tlačítka je labela, která informuje o stavu portu (Connected/ Not Connected). Další dvě tlačítka zahájí či ukončí měření.

V pravé části aplikace je možné měnit vzorkovací frekvenci a krok v grafu. Vždy je potřeba danou změnu potvrdit tlačítkem „Change“.

Pod grafem se nachází tlačítka pro uložení grafu do obrázku a export naměřených dat. Také je možné posouvat se v grafu pomocí horizontálního posuvníku. Vpravo dole se nachází ještě tlačítka pro reset měření, které vynuluje naměřená data, nastaví graf na začátek a také tlačítka pro manuální překreslení grafu (toto není v současné fázi příliš potřeba, tlačítka slouží spíše pro účely testování, proto bude v budoucnu odstraněno).

Popis os v grafu se vykresluje při vykreslování dat do obrázku. Hodnota časové osy (osa X) je proměnná, ale hodnota osy Y se nevypisuje. V tuto chvíli se do grafu vypisují přímo hodnoty z AD převodníku (0 až 1024, 10-bitové rozlišení). Do budoucna by bylo

možné hodnoty přepočítávat přímo na pH a EC, ale nyní by to bylo předčasné, protože u EC není ještě není rozhodnuto, jaká bude maximální měřená hodnota. U pH by to možné bylo, protože by se mělo pohybovat v rozmezí od 0 do 14.

8.3 Práce s porty

Práce se sériovými porty je v C# velmi pohodlná. Framework .NET obsahuje knihovnu IO.PORTS, která má objekt SerialPort, který implementuje většinu potřebných funkcí. Na začátku programu je deklarován objekt pro práci se sériovým portem:

```
SerialPort port = new SerialPort();
```

Tento port je samozřejmě potřeba inicializovat. Údaje pro komunikaci byly zmíněny výše v rámci kapitoly věnující se nastavení mikroprocesoru. Pro nastavení potřebných parametrů je v programu vytvořena následující funkce:

```
private bool initPort(string portName)
{
    port.PortName = portName;
    port.BaudRate = 38400;
    port.DataBits = 8;
    port.Parity = Parity.None;
    port.StopBits = StopBits.One;
    //this function is called when some data is received
    port.DataReceived += Port_DataReceived;

    try
    {
        port.Open();
        return true;
    }
    catch
    {
        System.Windows.MessageBox.Show("Can not open device!");
        return false;
    }
}
```

Funkce inicializuje port, který je předán funkci v parametru portName. Portu je potřeba přiřadit funkci, která se zavolá v případě, že dorazí na port nějaká data. Vyvolání funkce si již systém obslouží sám.

Tato funkce se volá až v případě, že uživatel vybere vhodný port ze seznamu. Tento seznam se naplní automaticky po startu aplikace nebo stiskem tlačítka „Get Ports“. Přitom se zavolá následující funkce:

```
public void getPorts()
{
    //this will actual detected ports in system
    ports = SerialPort.GetPortNames();

    cbPorts.Items.Clear(); //clear previous ports in combobox

    foreach (string s in ports) //this will add each port in list to combo box
    {
        cbPorts.Items.Add(s);
    }
    //if there is some ports in combobox, prechoose first one

    if (cbPorts.Items.Count > 0) cbPorts.SelectedIndex = 0;
}
```

„ports“ je proměnná, která je vytvořena také pro celou třídu hlavního okna a jedná se o pole string. Výše uvedená funkce získá ze systému všechny dostupné COM porty. Jsou vrácena

pouze jména ve zkráceném formátu („COM1“, apod.). V tomto formátu je potřeba zadat jméno portu do funkce `initPort`. Součástí funkce je i naplnění příslušného comboboxu (roletové menu). Prvně je nutné combobox vymazat, jinak by se při opakovaném stisku tlačítka do seznamu přidávaly další a další položky a uživatel by pak mohl vybrat port, který je již odpojený. Naštěstí pro tyto případy je zavedeno ošetření v rámci funkce pro inicializaci portu. Pokud se nepodaří z nějakého důvodu připojit k portu, tak je uživatel o této situaci informován a nedojde k pádu aplikace.

Poslední řádek ve funkci zjistí, jestli se podařilo načíst nějaké porty. Pokud je v seznamu více než jedna položka, tak řádek zobrazí automaticky v comboboxu první položku ze seznamu. Tento způsob přidá na uživatelském komfortu při ovládání aplikace, protože většinou pak stačí jen kliknout na tlačítko „Connect“ (jelikož daný port je již v comboboxu vybrán).

Pokud uživatel vybere konkrétní port a klikne na tlačítko „Connect“ je zavolána následující funkce, která se pokusí připojit port:

```
private void btnConnect_Click(object sender, RoutedEventArgs e)
{
    if (!port.IsOpen)
    {
        if (cbPorts.Items.Count > 0)
        {
            if (initPort(cbPorts.SelectedItem.ToString()))
            {
                lbConnectStatus.Content = "Connected";
                btnConnect.Content = "Disconnect";
            }
            else
            {
                lbConnectStatus.Content = "Not Connected";
            }
        }
    }
    else
    {
        port.Close();
        lbConnectStatus.Content = "Not Connected";
        btnConnect.Content = "Connect";
    }
}
```

Tato funkce prvně ověří, jestli už port není otevřený. V případě, že je, tak port uzavře. Jedná se totiž o situaci, kdy uživatel se již jednou připojil. O tomto stavu je informován změnou informačního textu a také změnou nápisu tlačítka, které se přepíná mezi „Connect“ a „Disconnect“, podle toho v jakém stavu je právě port a tedy co je možné očekávat za akci. Pokud není port otevřený a v seznamu portů je alespoň jedna položka, je získáno jméno aktuálně vybraného portu a zahájen pokus o inicializaci. Funkce `initPort` má návratovou hodnotu typu `bool` (vrací „true“ při povedeném pokusu o připojení a „false“ při nepřipojení k portu). Díky tomu je možné ji použít přímo v rámci „if“ příkazu, který rovnou pozná, jestli je port již připojený a podle toho přenastaví nadpis tlačítka a informačního textu.

Po připojení k portu je již možné na něj odesílat data. Pro účely poslání dat byla vytvořena následující funkce:

```
private void SendData(string data)
{
    if (!string.IsNullOrWhiteSpace(data))
    {
        if (port != null && port.IsOpen)
        {
            port.WriteLine(data);
        }
    }
}
```

```

    }
}
}

```

Funkci se předají data, ve formě řetězce string. Pokud jsou funkci předána nějaká data, tak se ověří, jestli port existuje a je otevřený. V případě, že je vše pozitivní, tak se zapíše na port jeden řádek s danými daty.

Nejsložitější částí práce s porty je funkce pro zpracování příchozích dat. Jedná se o funkci:

```
private void Port_DataReceived(object sender, SerialDataReceivedEventArgs e)
```

Tato funkce bude nyní rozebrána postupně. Na začátku je deklarována proměnná, do které se uloží příchozí zpráva. Celé zpracování zprávy je uzavřeno do try-catch bloku, aby bylo možné odchytil případnou výjimku. Přečtení zprávy se provede pomocí následujícího příkazu:

```
msg = port.ReadTo(";") + ";";
```

Objekt `SerialPort` má implementovanou funkci, která dokáže číst řetězec, dokud nenarazí na konkrétní znak. Toto je velmi užitečná funkce pro separaci dat, pokud jako v daném případě jsou zprávy ukončovány standardním znakem. Protože tento znak není již do zprávy zahrnut, tak je dobré jej ještě ke zprávě „ručně“ přidat, což se bude hodit při separaci hodnoty ze zprávy.

V první fázi je však ještě rozhodnuto, jestli se jedná o data z prvního nebo druhého kanálu:

```
Match m;
m = Regex.Match(msg, @"(00,[-\d]+);");
//if message is "00,0000;" to "00,9999;" it matches

```

Rozhodování se provádí pomocí regulárního výrazu. Pro používání regulárních výrazů je potřeba nezapomenout přidat správnou knihovnu (`using System.Text.RegularExpressions;`). Daný regulární výraz se uloží do `m.Success`, pokud přijde zpráva, která se skládá z 00 čárky čísla a středníku. Číslo může být libovolně dlouhé. V našem případě však přijde maximálně hodnota 1024 (AD převodník má 10 bitové rozlišení). Pokud je regulární výraz ve shodě s přijatou zprávou, pak se pokračuje následovně:

```
if (m.Success)
{
    lock (thisLock) //lock this content to prevent some crossthreading
    {
        //separate number from message and convert it to Int32
        pH_buffer[pH_actualBufferPosition] = Convert.ToInt32(msg.Split(',', ';')[1]);
        pH_actualBufferPosition++;
        if (pH_actualBufferPosition >= pH_buffer.Length) pH_actualBufferPosition = 0;
        newValuePH = true;
    }
}

```

Celý kód je uzamčen pomocí `lock` funkce. Je to potřeba z důvodu `crossthreadingu`, což v překladu znamená křížení vláken. Zpracování bufferu portu totiž běží v samostatném vlákně oproti hlavní aplikaci, která vlastní proměnné, ke kterým se ve funkci přistupuje. Pokud by k některé z proměnných chtěly přistoupit funkce, tak by to zároveň vedlo k výjimce a tím ke ztrátě dat či pádu aplikace.

Na začátku je rozseparována zpráva. V C# k tomuto účelu existuje velmi šikovná funkce `Split`, kterou má definovaná každá proměnná typu `string`. Tato funkce vytvoří pole, kde na 0. pozici jsou data před prvním separačním znakem, na 1. pozici jsou námi požadovaná data a na 2. pozici jsou data za druhým separačním znakem. Řetězec je ještě potřeba převést na číslo, což je zde řešeno integrovaným převodníkem z třídy `Convert`. Získaná hodnota je uložena do příslušného bufferu na aktuální pozici. Tato pozice je při každém uložení dat inkrementována. Pokud dojde buffer nakonec, tak se začne přepisovat od

začátku. Podobný systém funguje například u osciloskopů. Nakonec je ještě nutné nastavit proměnnou `newValueH`, která informuje hlavní vlákno o nově načtené hodnotě.

Separace dat z druhého kanálu funguje naprosto stejným stylem, takže zde již nebude rozváděna. Jediným rozdílem je, že se data uloží do bufferu `EC_buffer`.

Tímto byly popsány všechny funkce, které zajišťují komunikaci přes sériovou linku. Dále bude vysvětlena funkce týkající se spuštění měření, exportu a zobrazení dat.

8.4 Zahájení a ukončení měření

Pro zahájení a ukončení měření jsou v GUI tlačítka „Start Measurement“ a „Stop Measurement“. Tyto tlačítka pouze spustí nebo ukončí časovač, který posílá do procesoru znak „0“, kterým si vyžádá odeslání naměřených dat. Časovač je deklarovaný na začátku programu:

```
DispatcherTimer timer_getValue = new DispatcherTimer();
    Inicializace probíhá při načítání okna, ve funkci MainWindow():
//this timer sends request to mcu every sample frequency moment
timer_getValue.Tick += new EventHandler(timer_getValue_Tick);

if (actualSamplingFrequency < 1)
    timer_getValue.Interval = new TimeSpan(0, 0, 0, 0, (int)(actualSamplingFrequency *
1000)); //when sampling frequency is lower than 1s it is need to set it up as ms
else
    timer_getValue.Interval = new TimeSpan(0, 0, (int)actualSamplingFrequency); //for
setting sampling frequency in seconds
```

První řádek nastaví, která funkce se má zavolat v případě, kdy dojde k uplynutí nastavené doby časovače. Protože je možné v programu nastavovat vzorkovací frekvenci, je potřeba nastavit i daný interval. Pokud se jedná o interval, který je menší než 1s, pak je potřeba hodnotu přepočítat na milisekundy, v opačném případě se nastaví čas přímo v sekundách.

Obslužná funkce pro tlačítko „Start Measurement“ pouze zavolá následující funkci, která spustí časovač:

```
timer_getValue.Start(); //Start data acquisition timer
```

Obslužná funkce pro tlačítko „Stop Measurement“ zavolá funkci, která vypne časovač:

```
timer_getValue.Stop(); //Stop data acquisition timer
```

8.5 Export dat

Aby bylo možné dále s naměřenými daty pracovat a provádět pokročilejší analýzy bylo potřeba zajistit možnost exportu dat. Toto má na starosti funkce pro obsluhu tlačítka `btnExportData_Click`. Funkce vyvolá `SaveFileDialog`, pomocí kterého uživatel nadefinuje název souboru, do kterého se má provést výpis.

```
Microsoft.Win32.SaveFileDialog saveFileDialog = new Microsoft.Win32.SaveFileDialog();

saveFileDialog.Filter = "Text files (.txt)|*.txt"; //File dialog will display only
images with txt suffix
saveFileDialog.FileName = DateTime.Today.ToShortDateString().Replace('.', '_') +
"pH_and_EC_data.txt"; //preset name of file
if (saveFileDialog.ShowDialog() == true) //display dialog and if exit with "OK"
button, save image
{
    if (saveFileDialog.FileName != string.Empty)
    {
```

Aby se zjednodušila práce se soubory, má uživatel předdefinovaný název souboru. Také je nastaven filtr pro `SaveFileDialog`, aby zobrazoval jen soubory s koncovkou `txt`.

Pro jednoduchou možnost importu dat do tabulkových procesorů byl zvolen zápis dat v následujícím formátu:

```
Pozice_hodnoty_v_bufferu;hodnota_pH;hodnota_EC;
```

Jednotlivé položky jsou odděleny středníkem, který je možné nastavit například v MS Excel jako oddělovač a MS Excel si sám data rozdělí do sloupců.

```
//if user choosed some filename
//create new file based on selected name of file
using (StreamWriter sw = new StreamWriter(saveFileDialog.FileName, false))
{
    int countOfZeros = 0;
    //this will count when data will be zero and end instead of write zeros to the file

    for(int i = 0; i<bufferLength; i++) //go through whole buffer
    {
        sw.WriteLine(i.ToString() + ";" + pH_buffer[i].ToString() + ";" +
EC_buffer[i].ToString() + ";"); //write one line with separator character

        if (pH_buffer[i] != 0 && countOfZeros != 0) countOfZeros--;
        if (pH_buffer[i] == 0) countOfZeros++;
        //if there are ten zero values behind, stop writing
        if (countOfZeros >= 10) break;
    }

    sw.Close(); //close the stream
```

Funkce je uzavřena v using statement, aby bylo zajištěno vyčištění paměti po dokončení operace. V první části se vytvoří stream pro zápis do souboru. Název souboru se získá z SaveFile Dialogu, kde uživatel zvolil cestu i název souboru. Na základě délky bufferu se projdou všechny prvky a zapíší se vždy na samostatný řádek spolu s pozicí v bufferu. Během zápisu se hlídá, jestli již nejsou hodnoty nulové. Pokud je za sebou 10 hodnot nulových, tak se zápis ukončí, aby se zbytečně nezapisovaly nuly. Po dokončení zápisu je potřeba ještě uzavřít soubor. Uvolnění streamu z paměti již zajistí blok using.

8.6 Vykreslování grafu

Vykreslování grafu je nejobtížnější částí programu. Navíc se jedná o část, kterou bude potřeba ještě více rozšířit a dopracovat, čemuž bude věnována pozornost primárně v rámci navazujícího pokračování vývoje během tvorby praktické maturitní práce. Nyní je možné do grafu vykreslit data načtená v bufferu. Je možné si změnit vzorkovací frekvenci, na které závisí také vykreslování v grafu.

Vykreslení grafu probíhá vždy, když se nastaví flag newValuePH nebo newValueEC. Většinou jsou nastaveny oba, protože data z procesoru přijdou vždy pro oba kanály. Aktuální stav flagů ověřuje časovač, který má nastavenou frekvenci 2Hz. Při uplynutí doby se vyvolá funkce dispatcherTimer_Tick. V této funkci se zkontroluje, jestli došlo k načtení nových hodnot a poté se spustí funkce drawGraph().

Ve funkci drawGraph() se vykreslí spojnice bodů do grafu. Práce s grafikou probíhá na vektorové úrovni. Pro práci s vektorovou grafikou slouží v C# WPF objekt Canvas.

V programu je pro něj vytvořena instance canGraph.

```
canGraph.Children.Clear(); //clean previous graph
createAxisLines(); //draw axis lines
renderTextToCanvas(15, 10, "pH or EC[%]", Color.FromRgb(0, 0, 0), canGraph);
renderTextToCanvas(canGraph.Width - 50, canGraph.Height - 35, "time[" +
actualSamplingFrequency.ToString() + "s]", Color.FromRgb(0, 0, 0), canGraph);
```


Na začátku funkce je potřeba prvně vyčistit celý objekt canGraph. Poté se vytvoří osy podle zvoleného kroku a k osám se vykreslí i popis os. Funkce pro vykreslení os jsou převzaty z tutoriálu (13). Funkci pro vykreslení textu bude popsána níže.

```
//update graphLines and clean the flags
if (newValuePH)
{
    pH_graphLine = createPolyLine(pH_buffer, true); //create line based on actual data
in buffer
    newValuePH = false;
    canGraph.Children.Add(pH_graphLine);
}
if (newValueEC)
{
    EC_graphLine = createPolyLine(EC_buffer, false); //create line based on actual
data in buffer
    newValueEC = false;
    canGraph.Children.Add(EC_graphLine);
}
```

V další části funkce se naplní objekt typu Polyline (pH_graphLine a EC_graphLine), který se přidá do objektu canGraph. Polyline je velmi užitečný objekt vektorové grafiky ve WPF. Podle přiřazené kolekce bodů se automaticky vytvoří jejich spojnice. Pro vytvoření polyline na základě dat z bufferu byla vytvořena samostatná funkce. Funkce byla při své tvorbě inspirována podobným systémem, který je v návodu, z něhož vycházela funkce pro vykreslení os. Daná funkce musela být upravena tak, aby vytvořila data na základě předaného pole hodnot int. Funkci se také předává flag, jestli se jedná o pH (true) nebo EC(false). Tento flag rozhodne o barvě vytvořené čáry.

```
Polyline line = new Polyline();

Brush[] brushes = { Brushes.Red, Brushes.Green };

PointCollection points = new PointCollection();
int x_element = settedGraphStart;
```

Na začátku funkce se vytvoří objekt Polyline, který bude sloužit pro navrácení hodnoty z funkce. Vytvoří se také pole štětců, které definují barvu čáry. Dále se vytvoří kolekce bodů, do které se přidají všechny hodnoty bodů, které budou tvořit danou čáru. Jako poslední se nadefinuje hodnota, kde vykreslování grafu začíná. V aplikaci je totiž možné si pomocí posuvníku graf procházet, ve chvíli, kdy se již nevhodí na jednu obrazovku.

```
for (double x = xmin; x <= xmax; x += step)
{
    int yvalue = (int)ymax - buffer[x_element];
    if (yvalue < 0) yvalue = 0;
    points.Add(new Point(x, yvalue));
    x_element++;
}
```

Ve funkci je for cyklus, který vytvoří konkrétní bod a přidá jej do kolekce. V první fázi je potřeba invertovat Y osu, protože aplikace ve WPF mají bod 0,0 v levém horním rohu. Nicméně graf potřebujeme mít tak, aby měl 0. bod ve spodní části. Je samozřejmostí kontrolovat, aby se nevykreslil bod se zápornou hodnotou. Poté se již jen přidá bod do kolekce. Pozice X se inkrementuje podle nastaveného kroku. Jakmile je vytvořena kolekce bodů, tak stačí nastavit tloušťku čáry, barvu, přidat kolekci bodů a vrátit z funkce vytvořenou čáru:

```
line.StrokeThickness = 1;
if (pH) line.Stroke = brushes[0]; //pH line is red
if (!pH) line.Stroke = brushes[1]; //EC line is green
line.Points = points;

return line;
```


Z pohledu vykreslování grafu je další důležitou funkcí změna kroku v grafu. Jedná se o funkci pro obsluhu tlačítka `btnChangeNOS_Click`. Zde se podle položky v comboboxu s definovaným krokem překonvertuje krok do proměnné `step`:

```
step = Convert.ToDouble(cbNumberOfSamples.Items[cbNumberOfSamples.SelectedIndex]);  
//change steps
```

Poté se nastaví maximální možná hodnota posuvníku:

```
sbPosition.Maximum = EC_actualBufferPosition;
```

Na konec se povolí překreslení nastavením flagů `newValueEC` a `pH`, aby došlo k překreslení grafu.

Poslední funkcí, která ovlivní grafiku a stojí za zmínku, je řešení posouvání v grafu. Jedná se o funkci pro obsluhu eventu pro scrollování posuvníku `sbPosition_Scroll`. V této funkci se jen nastaví hodnota `settedGraphStart`, která se používá při vytváření spojnic bodů. Ve funkci se získá aktuální hodnota posuvníku a ta se pak načte do proměnné:

```
settedGraphStart = Convert.ToInt32(sbPosition.Value);
```

Poté opět dojde k nastavení flagů `newValueEC` a `pH`, aby došlo k překreslení grafu.

8.7 Uložení grafu do obrázku

Výše byla zmíněná funkce, kterou je možné vyexportovat data do txt souboru. Protože ne vždy je potřeba exportovat data, ale je vhodnější předat jen graf, bylo logickým krokem, přidat do aplikace možnost exportu grafu ve formě obrázku. Protože je graf vykreslen formou vektorové grafiky, bylo potřeba vytvořit funkci pro převod na obrázek. Jedná se o celkem častou operaci, kterou řešila již řada programátorů, tudíž se v našem případě vyzkoušely tři různé metody, jak objekt `Canvas` převést do obrázku. Uvedená funkce pochází z (14). Bylo potřeba do ní udělat zásah a to v části pro renderování do bitmapy. Zde se uvažuje pouze s rozměry objektu `Canvas`. Nicméně pokud je tento objekt v okně posunutý, pak je potřeba tento posun přičíst, jinak by došlo k oříznutí obrázku.

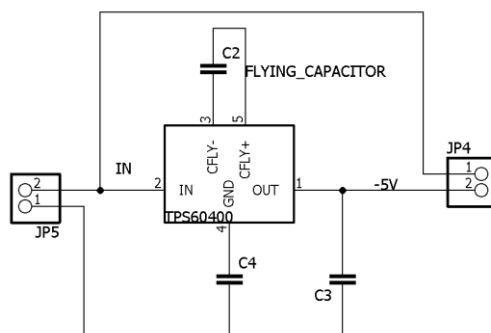
Obrázek je ukládán ve formátu `png`. Jedná se o bitmapová data s kompresí, nicméně bezztrátovou. Oproti `bmp` má obrázek přibližně poloviční velikost a přitom kvalita obrazu je stejná.

9 Realizace EC převodníku

9.1 Teoretická část

9.1.1 Napájecí obvod

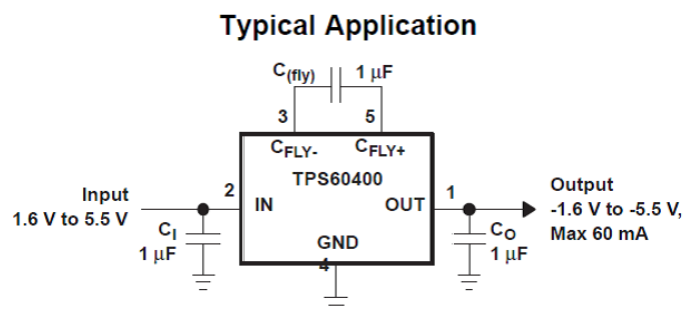
Na napájení EC převodníku je potřeba kladné i záporné napětí z důvodu, aby rozsah zesílení operačních zesilovačů mohl být 0 V až 5 V. K tomuto byl použit tzv. charge pump neboli nábojová pumpa, která umožňuje invertovat vstupní napětí (viz kapitola 1.1.1.1 *Charge pump neboli nábojová pumpa*). Vstupní napětí v případě realizovaného EC převodníku bylo tedy 5 V a výstupní -5 V. Tento napájecí obvod je využit i u obvodu pro měření pH hodnot.



Obrázek 12 - schéma napájecího obvodu

Vybraná a osazená charge pump TPS60400 od firmy Texas Instruments je z řady zařízení, jenž umí invertovat vstupní napětí. Mimo jiné v sobě obsahuje integrovanou Shottkyho diodu, která bývá potřeba pro uvedení zařízení do provozu.

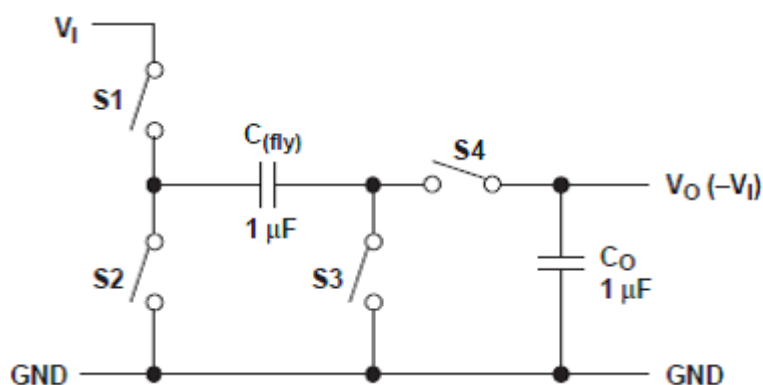
Kondenzátory byly vybrány s co nejmenší hodnotou ESR (neboli zdánlivý odpor), aby bylo dosaženo co nejnižšího výstupního odporu.



Obrázek 13 - nábojová pumpa(15)

9.1.1.1 Charge pump neboli nábojová pumpa

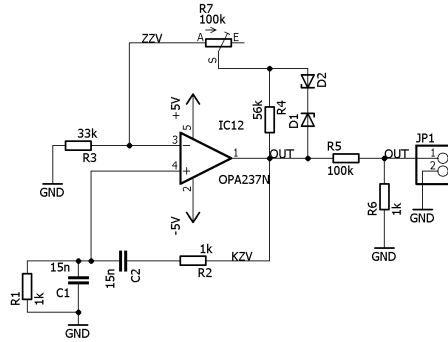
Nábojová pumpa funguje jako DC-DC měnič. Ke své činnosti nevyžaduje cívku, pouze keramické či elektrolytické kondenzátory. Z detailního popisu a schématu z datasheetu(15) součástky, lze zjistit, že při prvním půl cyklu je spínač S1 a S3 sepnutý a kondenzátor C (fly) se nabíjí ze vstupního napětí V_I . Při druhém půl cyklu jsou spínače S1 a S3 rozepnuty a spínače S2 a S4 sepnuty, to znamená, že kladný výstup je sepnutý na zem a záporný výstup je sepnutý na výstup pumpy. Při druhém půl cyklu se nabíjí kondenzátor C_O , který se při prvním půl cyklu vybíjí do zátěže. Nábojové pumpy mohou fungovat jako invertující, zesilující a snižující napětí. Jejich výhodou je ve velmi malém provedení a možnosti osadit běžné kondenzátory namísto cívky. Nevýhodou je však ve velmi nízkém odběru (maximálně až 300 mA) a také v nemožnosti skokové změny napětí(16).



Obrázek 14 - detailní popis nábojové pumpy

9.1.2 Oscilátor

Jak již bylo zmíněno v předchozích kapitolách, EC sondu je potřeba napájet střídavým proudem. Z tohoto důvodu bylo nutné vytvořit oscilátor, který bude oscilovat na takové frekvenci, aby se zmenšil vliv parazitní kapacity na elektrodách sondy. Jelikož bylo k další práci se signálem nutné použít dva operační zesilovače, bylo využito čtyř operačních zesilovačů v jednom pouzdře. Z jednoho operačního zesilovače pak byl udělán oscilátor. Jak již bylo vysvětleno v předchozí podkapitole, byla použita nábojová pumpa, která invertovala vstupní napětí z 5 V na -5 V, což znamená, že zem v tomto případě měla napětí -5 V.



Obrázek 15 - schéma oscilátoru

Součástka, která byla poskytnuta firmou ELEDUS s.r.o. byla dostačující. Jednalo se o TL084CD (17).

Tento operační zesilovač má na vstupu JFET tranzistory, tzn., že jeho vstupní odpor je vysoký, a tudíž má nízký vstupní proud, z čehož plyne, že nezatěžuje zdroj signálu a ve výsledku tak nezkrsluje signál posílaný EC sondou (vyšší přesnost měření).

Jedná se o zapojení takzvaného oscilátoru s Wienovým článkem, který dokáže vytvořit sinusový signál bez fázového posunu.

Výstupní frekvence se získá díky výpočtu (6) a výstupní napětí pomocí změny zenerových diod D1 a D2, které v tomto případě byly zvoleny tak, aby výstupní napětí bylo 5V. (18)

$$F_o = \frac{1}{2\pi RC} \quad (7)$$

Výstupní frekvence byla nastavena na 10 kHz, aby se netvořila nežádoucí kapacita na elektrodách EC sondy. Frekvence však nebyla přesná. Pro přesnou frekvenci by byly potřeba přesnější rezistory a kondenzátory např. svitkové. (19)

9.1.3 Zpracování signálu

Další operační zesilovač byl využit pro zesílení signálu (je zapojený jako neinvertující). Jeho zesílení A_U se určuje podle výpočtu (7). R_2 je v tomto případě 6,8 k Ω , vstupní napětí U_{vstup} se rovná 195 mV, R_1 se mění podle hodnoty EC, což znamená že R_1 je nahrazený EC sondou, která mění svůj odpor podle hodnoty EC. Pokud se změří EC například o hodnotě 4000 μ S/cm, tak se zjistí, že na výstupu zesilovače je napětí rovno cca 6 V. S těmito hodnotami je možno spočítat odpor EC sondy. Použité hodnoty pro výpočet byly měřeny při střídavém napětí peek-to-peek.

$$A_U = \frac{U_{výstup}}{U_{vstup}} = \frac{R_1 + R_2}{R_1} \quad (8)$$

$$A_U = \frac{6 V}{195 mV} = \frac{R_1 + 6,8 k\Omega}{R_1} \quad (9)$$

$$A_U = \frac{6 V}{195 mV} \cong 32 \quad (10)$$

$$R_1 = \frac{6800}{31} \quad (11)$$

$$R_1 = 220\Omega \quad (12)$$

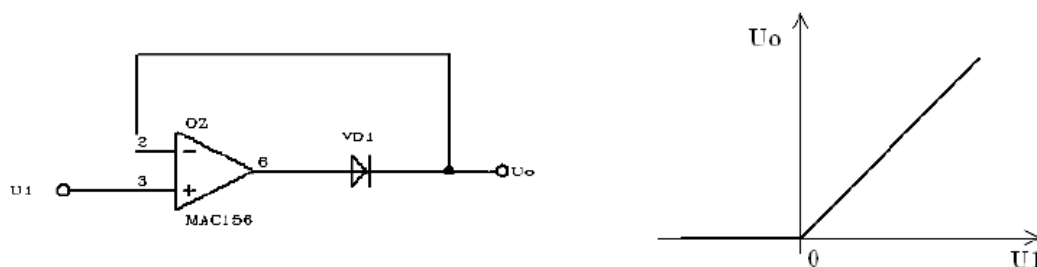
Pro zpětnou kontrolu stačí z odporu vypočítat zpět vodivost tj. $R_1^{-1} \cong 4 \text{ mS/cm}$.

9.1.4 Usměrnění signálu

Výstup po úpravě signálu z EC sondy bylo potřeba ještě usměrnit, to znamená zbavit se záporné vlny napětí. První pokus o usměrnění byl s použitím klasického dvoucestného usměrňovače neboli Graetzového můstku se zapojením, u kterého bylo možno měnit offset neboli posunutí signálu a jeho rozsah. Dané řešení bylo nalezeno na stránkách (20) a bylo použito a upraveno podle potřeb této práce. Bohužel zde byl nevyhovující způsob usměrnění, jelikož vznikaly problémy s nastavováním jak offsetu, tak i rozsahu. Proto byl nakonec pro usměrnění signálu využit operační zesilovač zapojený jako ideální dioda.

9.1.4.1 Operační zesilovač jako usměrňovač

Zapojení operačního zesilovače jako ideální diody, je používáno v obvodech, kde se nachází nízké napětí a obyčejné usměrňovací diody neposkytují takovou linearitu. V našem případě bylo nutné využít rychlé diody, jelikož usměrňují signál, který kmitá na 10 kHz.(jednocestné)



Obrázek 16 - ideální dioda

9.1.5 Ochrana ADC

Před tím než je výsledný usměrněný signál připojen na piny ADC převodníku, je vhodné použít ochranné diody, aby nedošlo k poškození samotného procesoru vyšším napětím. Ty se nacházejí za RC článkem.

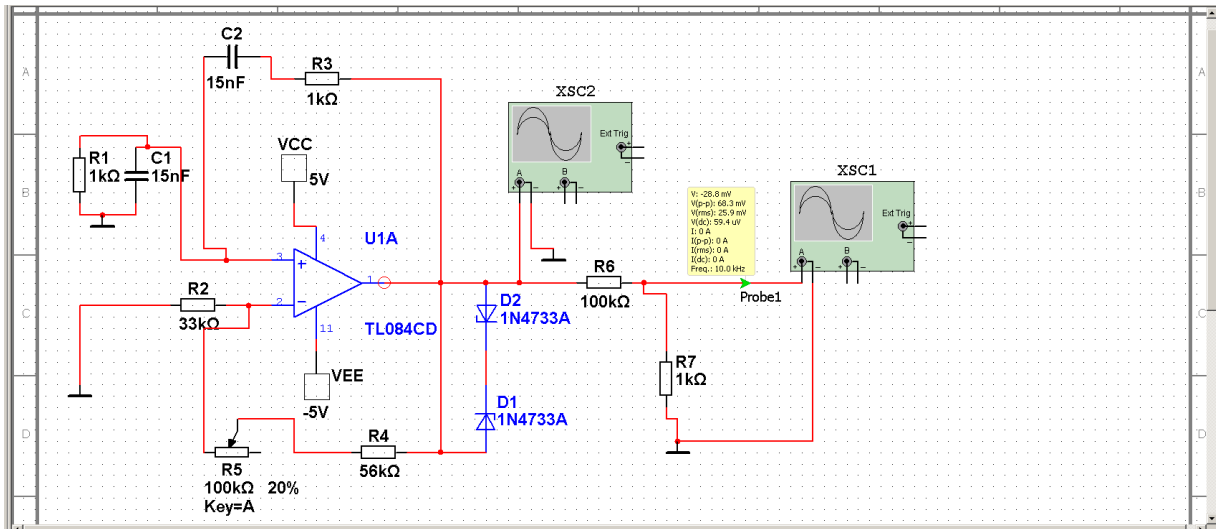
9.1.6 Důvod použití TL084CD

Jak již bylo zmíněno, operační zesilovač TL084CD má JFET tranzistory na vstupech a proto je jeho spotřební proud velmi nízký. Je cenově dostupný, jeho cena se pohybuje kolem 8 Kč. Je to součástka, která je ve firmě ELEDUS s.r.o. běžně dostupná a proto byla využita i v tomto případě.

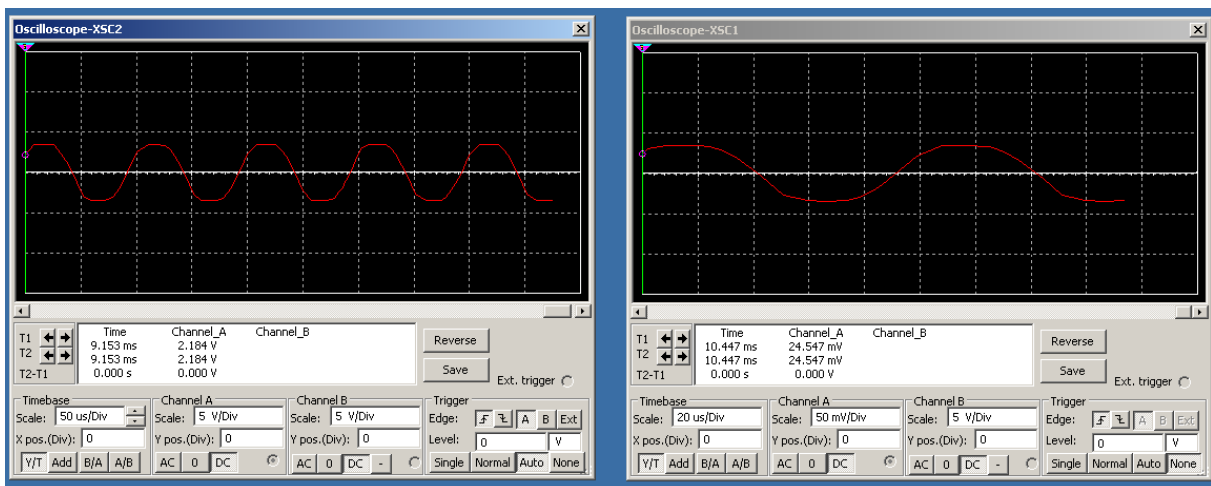
9.2 Praktická část

9.2.1 Simulace

Před začátkem navrhování a vytvoření DPS, byl nasimulován obvod v softwaru NI Multisim. Simulace umožnila zkontrolovat chod obvodu. Jelikož obsahuje i osciloskop bylo možné zjistit, jak daná část obvodu funguje. Muselo se však počítat s mírnou odchylkou, protože simulace neobsahuje reálné chování součástek, jejich zdánlivý odpor či kapacitu.



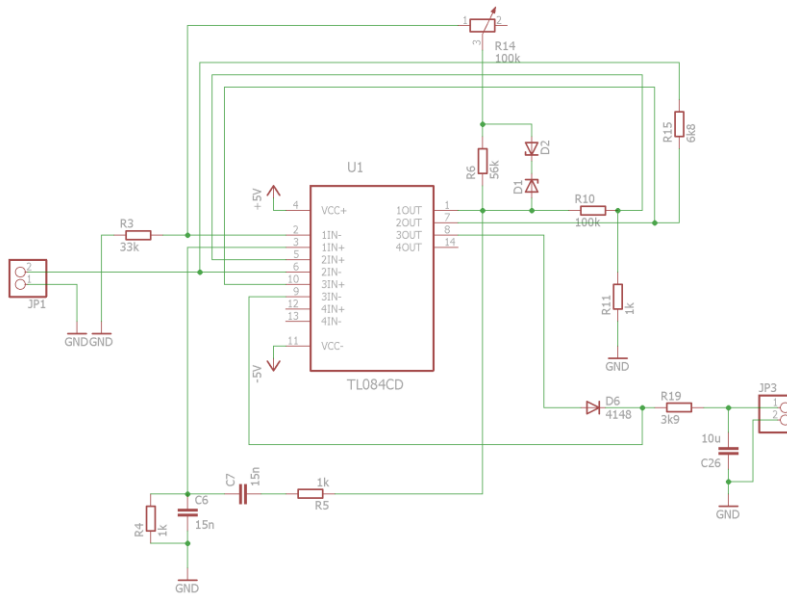
Obrázek 17 - simulace oscilátoru



Obrázek 18 - sinusová křivka před i za rezistorem

9.2.2 Návrh a vytvoření desky

Prvním nutným krokem bylo navrhnutí schématu a desky v softwaru EAGLE 7.3. Free.



Obrázek 19 – schéma EC desky

Při navrhování bylo maximálně využito snahy o použití součástek, které pro účely dané práce poskytla firma ELEDUS. K vyleptání navržené desky byla využita firemní leptací stanice.

Jako první bylo nutné osvětlit cuprexitivovou desku, která se poté nechala vyleptat.

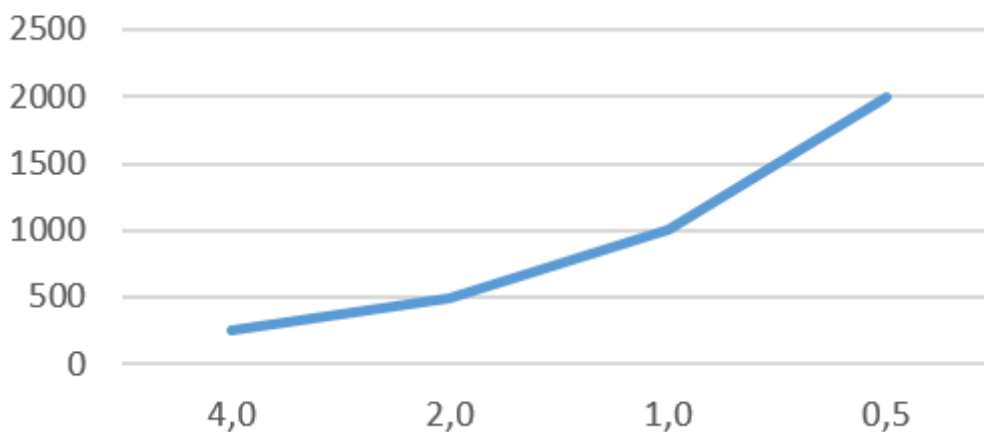
Po vyleptání mohla být deska osazena všemi potřebnými součástkami. Většina z nich byla v SMD pouzdře. Některé však v SMD provedení nebyly, proto bylo nutné do desky vrtat.

Po tomto procesu jsem již bylo možné měřit hodnoty EC.

9.2.3 Měření EC

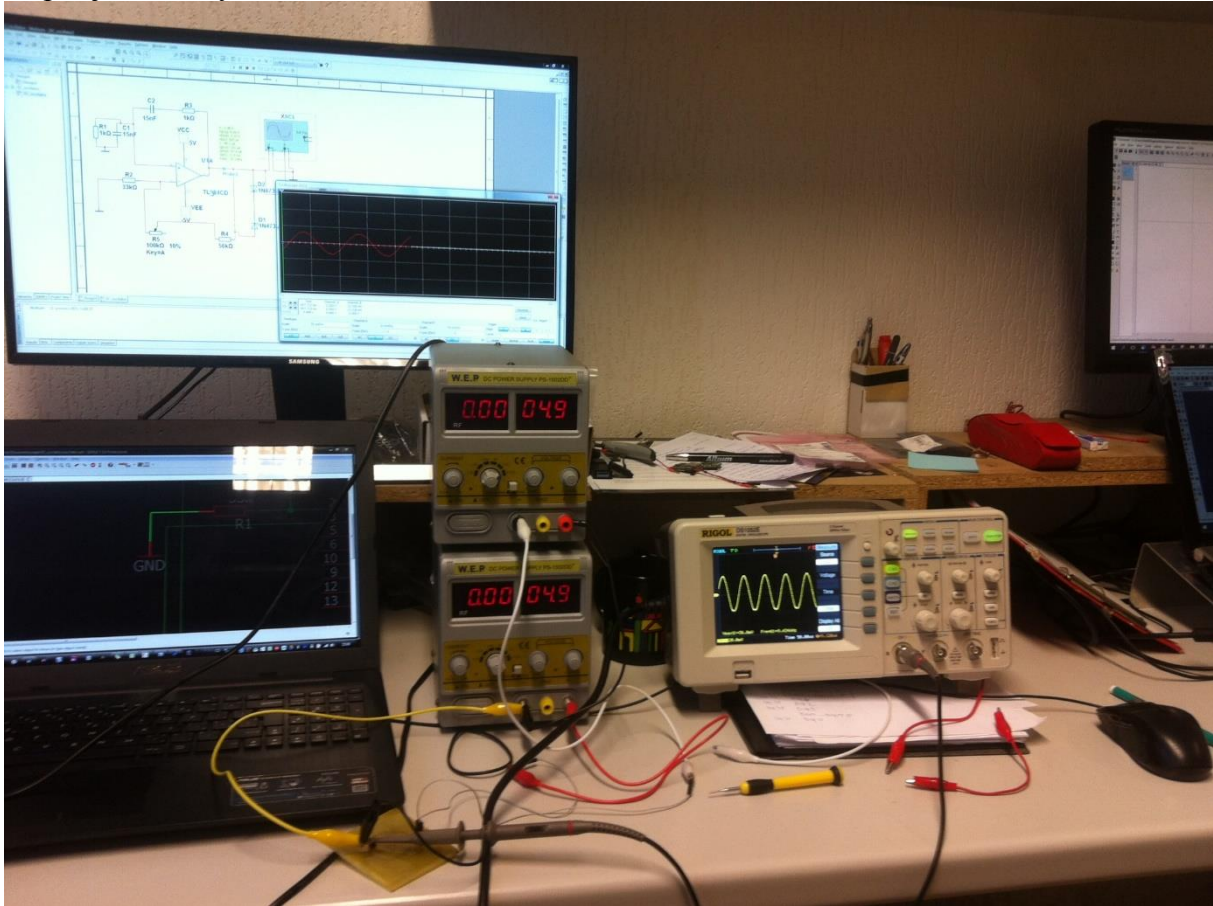
Měření EC bylo prováděno připojením sondy k elektronice a následném připojování sondy osciloskopu k elektronice, který byl k tomuto účelu propůjčen, na výstup druhého a potom i třetího zesilovače, aby bylo možné sledovat, jak se mění hodnota výstupního napětí při měření EC. Pro kontrolu a případnou kalibraci elektroniky byl využit komerční EC metr, s nímž byly porovnávány měřené hodnoty a tím se ověřila správná funkce vytvořeného EC metru.

Odpor Sondy



Obrázek 20 - graf měření EC

Odpor je uvedený v tisících, hodnota EC v S/cm.



Obrázek 21- zkouška oscilátoru

10 Realizace pH převodníku

10.1 Teoretická část

10.1.1 Napájecí obvod

K napájení pH převodníku bylo nutné použít 5 V až -5 V stejně jak tomu je u EC převodníku. Z tohoto důvodu byla použita nábojová pumpa, která je zmíněna u realizace EC převodníku.

10.1.2 Operační zesilovač

Sonda pro měření pH má velmi vysokou impedanci (odpor), proto jsem potřeboval použít operační zesilovač s velmi nízkým vstupním proudem v rozmezí stovek fA až jednotek pA.

Použil jsem operační zesilovač LM7702(21). Tento zesilovač sice nepatří mezi nejlevnější, stojí kolem 100 Kč, ale má výhodu v Rail to Rail vstupech a výstupech, jeho vstupní proud je +/- 200 fA. Jeho pouzdro obsahuje dva operační zesilovače. Jeden využiji pro zesílení signálu ze sondy a druhý pro srovnání napětí zesíleného signálu sondy a referenčním napětím.

10.1.3 Zesílení výstupního signálu sondy

Aby bylo možné změnu hodnoty pH vůbec rozpoznat, je nutné zesílit výstupní signál sondy. Sonden měřící pH se řadí mezi sondy s velmi vysokým odporem. Operační zesilovač bylo tudíž nutné zapojit jako neinvertující, aby se zesílilo vstupní napětí. Poměr zesílení lze vypočítat pomocí vzorce (12). Zesílení je vypočítáno tak, aby mělo 18 kroků na jednotku pH (jeden krok je asi 5 mV).

$$U_{výstup} = U_{vstupní} * \left(1 + \frac{R_2}{R_1}\right) \quad (13)$$

(obr. Schéma zesilujícího OZ)

10.1.4 Referenční napětí pro diferenciální zesilovač

Pro měření diferenciálního zesilovače je nutné vytvořit referenční napětí, které nebude pro piny ADC na procesoru ATmega8 nebezpečné. K tomuto byl využit napěťový regulátor přesněji regulátor SPX5205 s výstupním napětím 3,3 V. Pro vyšší stabilitu referenčního napětí je dobré použít zdroj retenčního napětí, ten je však dražší. Z tohoto důvodu byl použit regulátor, který je navíc pro vývoj dostačující. Toto napětí se ještě upravuje napěťovým děličem, kterým se získává referenční napětí podle výpočtu (13). Tedy napětí o hodnotě 2,5 V.

$$U_{výstup} = \frac{R_2}{R_1 + R_2} * U_{vstup} \quad (14)$$

(obr. Regulátor + napěťový dělič)

10.1.5 Diferenciální zesilovač

Druhý operační zesilovač integrovaný v pouzdře byl použit pro měření rozdílu mezi referenčním napětím a zesíleným napětím výstupního signálu pH sondy. Z tohoto důvodu byl zapojen operační zesilovač jako diferenciální. Jeho výstupní napětí je dáno rozdílem vstupních napětí a zesílením operačního zesilovače. Tento operační zesilovač musel umožňovat využití funkce Rail to Rail, a proto jej bylo také nutné napájet +5 V a -5 V. Tato funkce dovoluje využít celého rozsahu zesílení čili 0-5 V.

(obr. Diferenčního zesilovače)

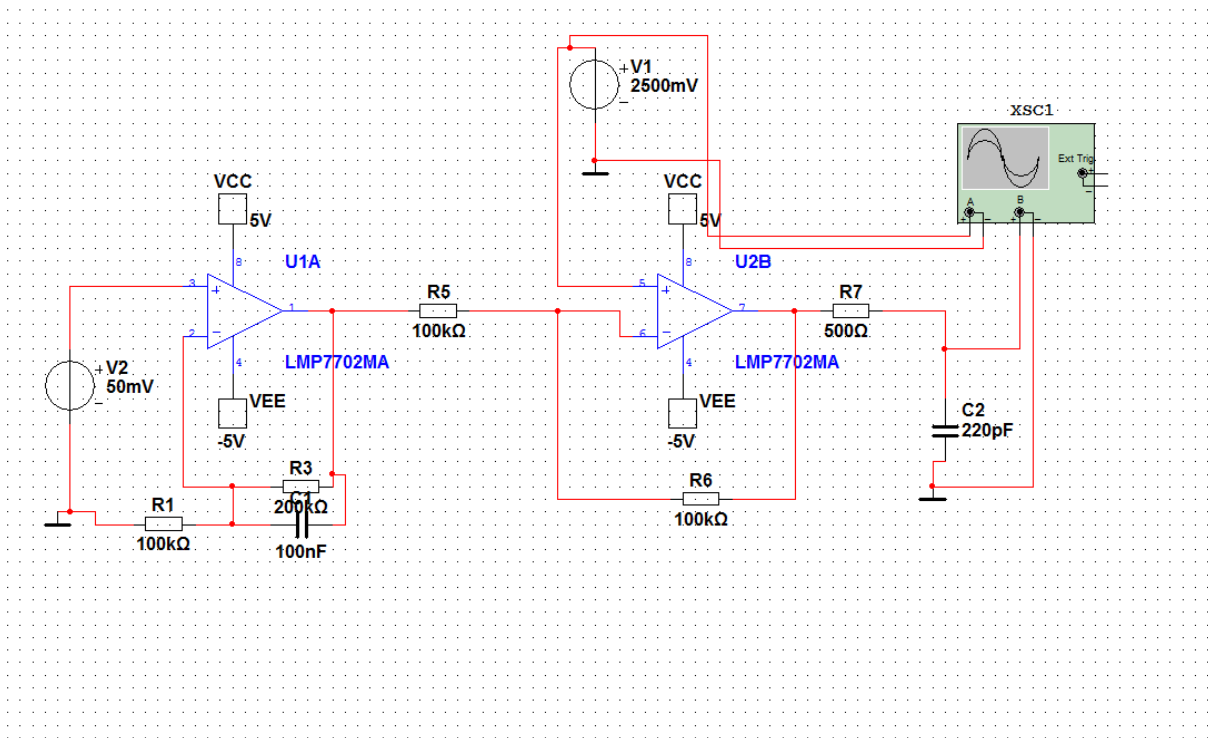
10.1.6 Výsledný signál

Za diferenčním zesilovačem se ještě nachází RC článek, který tento signál filtruje. Tento signál je dále pouštěn na piny ADC převodníku procesoru ATmega8.

10.2 Praktická část

10.2.1 Simulace

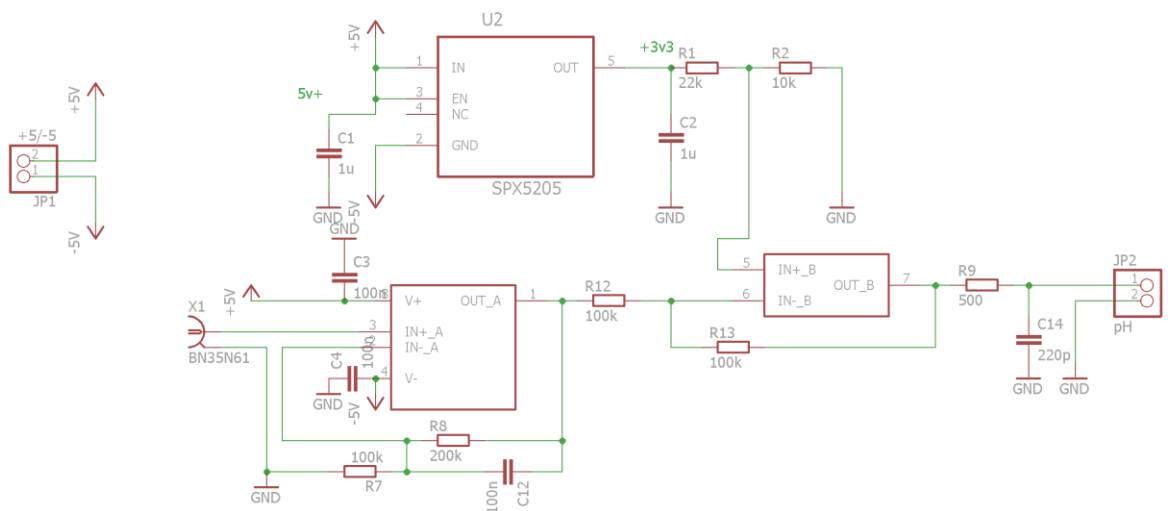
Před začátkem navrhování a vytváření DPS, byl nasimulován obvod v softwaru NI Multisim. Simulace umožnila zkontrolovat funkčnost obvodu, a jelikož obsahuje i osciloskop bylo možné zjistit, jak daná část obvodu funguje. Je však nutné počítat s mírnou odchylkou, protože simulace neobsahuje reálné chování součástek, jejich zdánlivý odpor či kapacitu.



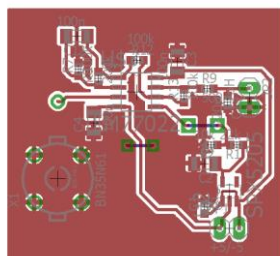
Obrázek 22 - simulace pH převodníku

10.2.2 Návrh a vytvoření desky

Prvním krokem bylo navržnutí schématu a desky v softwaru EAGLE 7.3. Free.



Obrázek 23 - schéma pH převodníku



Obrázek 24 - deska pH převodníku

Při navrhování bylo opět z ekonomických důvodů využito co nejvíce součástek, které poskytla firmou ELEDUS s.r.o. Po navrhnutí desky byla opět využita firemní leptací stanice.

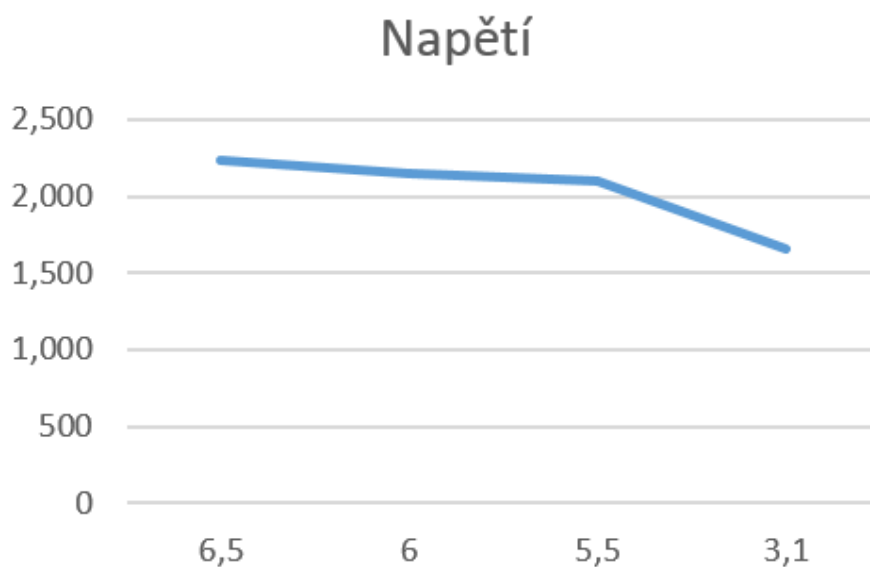
V první fázi je nutné osvětlit cuprexitovou desku a poté je možné ji vyleptat.

Po vyleptání byly osazeny všechny potřebné součástky. Většina jich byla v SMD pouzdře, některé však v SMD provedení nebyly, proto bylo nutné do desky vrtat.

Po tomto procesu již bylo možné zapojit pH sondu a vyzkoušet měření v praxi.

10.2.3 Měření pH

Prvním krokem bylo zkalibrování sondy a elektroniky pomocí pufrů neboli kalibračních roztoků s hodnotami pH 4 a 7. Při tomto kalibrování se zapsaly výstupní hodnoty napětí. Poté již bylo možné stanovit, jak probíhá změna napětí oproti změně hodnoty pH. Zjistilo se, že při dané elektronice je 18 kroků na jednotku pH a že jeden krok je cca 5 mV. Měření samozřejmě bylo porovnáno s komerčním pH metrem, díky kterému se zjistily malé odchylky z důvodu vyššího rozsahu měřiče.



Obrázek 25 - graf naměřených hodnot

Závěr

Realizace pH a EC metru mi přinesla spoustu nových poznatků v oblasti elektrotechniky, automatizace, programování mikrokontrolérů ale i chemie. Výsledné zařízení neměří nejpřesněji za to měří přesněji, než bylo předpokládáno, bude využito v dalších projektech a bude se dále zdokonalovat ve svých nedostatcích.

Pomocí zařízení bude možno měřit hodnoty ve spoustě automatizovaných systémech jako je hydroponie či akvaristika. Z měření byly sestaveny grafy a byly srovnány s komerčními zařízeními.

Získal jsem nové zkušenosti se senzory a s elektronikou a nové znalosti týkající se automatizace.

V budoucnu by mohl mít přesnější sondy a mohl by obsahovat přesnější komponenty pro analýzu signálů.

V práci jsem popsal vše potřebné pro pochopení problematiky měření pH i elektrické vodivosti v roztocích, a i fyzické principy, na nichž má práce zakládá.

Literatura

- [1] ŠIMURDA, Jiří a Milan RYCHTARIK. Měření pH [offline]. Neznámé, Neznámý [cit. 2016-02-11]. Dostupné z: Neznámá
- [2] MAŠLÁŇ, Bc. Stanislav. MĚŘENÍ VODIVOSTI KAPALIN [offline]. Brno, 2010 [cit. 2016-02-11]. Dostupné z: Neznámá. Diplomová práce. VUT v Brně. Vedoucí práce Doc. Ing. PETR BENEŠ, Ph.D.
- [3] PH metr HP-07. Típa [online]. Opava: TIPA, spol. s r.o., Neznámý [cit. 2016-02-11]. Dostupné z: <http://www.tipa.eu/cz/ph-metr-hp-07/d-123255/>
- [4] Digitální pH metr Hanna Instruments HI 991001. Conrad [online]. Praha: Conrad Electronic Česká republika, s.r.o., Neznámý [cit. 2016-02-11]. Dostupné z: http://www.conrad.cz/digitalni-ph-metr-hanna-instruments-hi-991001-2-az-16-ph-5-az-105-c.k104731#utm_source=heureka&utm_medium=agregator&utm_campaign=2016&utm_content=104731
- [5] BNC Connector Electrode PH Probe Sensor For Aquarium PH Controller Meter. Aliexpress [online]. Čína: Vito V industry, Neznámý [cit. 2016-02-11]. Dostupné z: http://www.aliexpress.com/item/BNC-Connector-Electrode-PH-Probe-Sensor-For-Aquarium-PH-Controller-Meter/2041161507.html?spm=2114.01010208.3.1.SI14Nr&ws_ab_test=searchweb201556_2,searchweb201644_2_505_506_503_504_502_10001_10002_10016_10005_10006_10003_10004,searchweb201560_2,searchweb1451318400_-1,searchweb1451318411_6449&btsid=27fa0670-e64c-425e-aef2-dd9b0df5aa33
- [6] Conductivity meter sensor/CM - 230. Aliexpress [online]. Čína: Pacific mall, Neznámý [cit. 2016-02-11]. Dostupné z: http://www.aliexpress.com/item/high-temperature-on-line-conductivity-meter-sensor-CM-230-high-temperature-on-line-conductivity-meter/1735071149.html?spm=2114.01020208.3.195.TVNe4y&ws_ab_test=201556_2,201527_1_71_72_73_74_75,0_0
- [7] Measuring the Salinity of Water. Agriculture Victoria [online]. Austrálie: Agriculture Victoria, Neznámý [cit. 2016-02-11]. Dostupné z:

<http://agriculture.vic.gov.au/agriculture/farm-management/soil-and-water/salinity/measuring-the-salinity-of-water>

- [8] Trans4mind. Balancing Acid/Alkaline Foods [online]. Neznámé: Peter Shepherd, Neznámý [cit. 2016-02-11]. Dostupné z: <http://www.trans4mind.com/nutrition/pH.html>
- [9] WikiKnihy. Programujeme jednočipy [online]. Neznámé: FDominec, Neznámý [cit. 2016-02-12]. Dostupné z: https://cs.wikibooks.org/wiki/Programujeme_jedno%C4%8Dipy
- [10] Welcome to the Atmel Software Framework. Atmel [online]. Neznámé: Neznámé, Neznámý [cit. 2016-02-12]. Dostupné z: <http://www.atmel.com/webdoc/asf/pr01.html>
- [11] Atmel. Atmel [online]. Neznámé: Neznámé, Neznámý [cit. 2016-02-12]. Dostupné z: http://www.atmel.com/images/atmel-2486-8-bit-avr-microcontroller-atmega8_1_datasheet.pdf
- [12] Khazama AVR Programmer. Khazama [online]. Neznámé: Neznámé, Neznámý [cit. 2016-02-12]. Dostupné z: <http://khazama.com/project/programmer/>
- [13] Draw a graph in WPF and C#. CsharpHelper [online]. Neznámé: Rod Stephens, Neznámý [cit. 2016-02-12]. Dostupné z: <http://csharpHelper.com/blog/2014/09/draw-graph-wpf-c/>
- [14] Questions: Save canvas to bitmap. Stack overflow [online]. Neznámé: Neznámé, Neznámý [cit. 2016-02-12]. Dostupné z: <http://stackoverflow.com/questions/5851168/save-canvas-to-bitmap>
- [15] TPS6040x Unregulated 60-mA Charge Pump Voltage Inverter. Texas Instruments [online]. Neznámé: Texas Instruments, 2001 [cit. 2016-02-12]. Dostupné z: <http://www.ti.com/lit/ds/slvs324b/slvs324b.pdf>
- [16] POCHOBRADSKÝ, Jakub. LABORATORNÍ PŘÍPRAVKY DO PŘEDMĚTU NAPÁJENÍ ELEKTRONICKÝCH ZAŘÍZENÍ [online]. Brno, 2011 [cit. 2016-02-12]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=41850. Bakalářská práce. VUT v Brně. Vedoucí práce Ing. MICHAL KUBÍČEK, Ph.D.
- [17] General purpose JFET quad operational amplifiers: TL084, TL084A, TL084B. STMicroelectronics [online]. Neznámé: STMicroelectronics, 2013 [cit. 2016-02-12]. Dostupné z: <http://www.st.com/web/en/resource/technical/document/datasheet/CD00000493.pdf>
- [18] Sine wave. In: Wikipedia: the free encyclopedia [online]. San Francisco [CA]: Wikimedia Foundation, 2015 [cit. 2016-02-12]. Dostupné z: https://en.wikipedia.org/wiki/Sine_wave
- [19] Oscilátory. Dmaster.cz [online]. Neznámé: Neznámé, Neznámý [cit. 2016-02-12]. Dostupné z: <http://www.dmaster.wz.cz/teorie/oscilatory/oscilatory.htm#wien>
- [20] EC/TDS/PPM Meter On Limited Budget. Octiva [online]. Neznámé: Neznámé, 2007 [cit. 2016-02-12]. Dostupné z: <http://www.octiva.net/projects/ppm/>
- [21] LMP770x Precision, CMOS Input, RRIO, Wide Supply Range Amplifiers. Texas Instruments [online]. Neznámé: Neznámé, 2005 [cit. 2016-02-12]. Dostupné z: <http://www.ti.com/lit/ds/symlink/lmp7702.pdf>
- [22] Vaše kompletní laboratoř. Laboratorní technika a chemické potřeby [online]. Praha: Labor-Komplet s.r.o., Neznámý [cit. 2016-02-14]. Dostupné z: <http://www.laboratorni-potreby.cz/ph-indikatorove-papirky/indikatorove-papirky-vhodne-na-mereni-ph-moce/>

Seznam obrázků

OBRÁZEK 1 - VZOREC PRO VÝPOČET EC PŘI RŮZNÝCH TEPLOTÁCH(1)	7
OBRÁZEK 2 - PH INDIKÁTORY(21)	9
OBRÁZEK 3 - SKLENĚNÁ ELEKTRODA(1)	9
OBRÁZEK 4 - DRAŽŠÍ METR (4)	12
OBRÁZEK 5 - LEVNÝ METR (3)	12
OBRÁZEK 6 - SONDA PH(5)	13
OBRÁZEK 7 - EC SONDA(6)	14
OBRÁZEK 8 – VÝVOJOVÁ DESKA S ATMEGA8	18
OBRÁZEK 9 – SCHÉMA VÝVOJOVÉ DESKY S ATMEGOU8	19
OBRÁZEK 10 - USB TO UART ROZHRANÍ	20
OBRÁZEK 11 - GUI ROZHRANÍ	26
OBRÁZEK 12 - SCHÉMA NAPÁJECÍHO OBVODU	34
OBRÁZEK 13 - NÁBOJOVÁ PUMPA(15)	35
OBRÁZEK 14 - DETAILNÍ POPIS NÁBOJOVÉ PUMPY	35
OBRÁZEK 15 - SCHÉMA OSCILÁTORU	36
OBRÁZEK 16 - IDEÁLNÍ DIODA	37
OBRÁZEK 17 - SIMULACE OSCILÁTORU	38
OBRÁZEK 18 - SINUSOVÁ KŘIVKA PŘED I ZA REZISTOREM	38
OBRÁZEK 19 – SCHÉMA EC DESKY	39
OBRÁZEK 20 - GRAF MĚŘENÍ EC	39
OBRÁZEK 21- ZKOUŠKA OSCILÁTORU	40
OBRÁZEK 22 - SIMULACE PH PŘEVODNÍKU	42
OBRÁZEK 23 - SCHÉMA PH PŘEVODNÍKU	42
OBRÁZEK 24 - DESKA PH PŘEVODNÍKU	43
OBRÁZEK 25 - GRAF NAMĚŘENÝCH HODNOT	43

Seznam vzorců

Vzorec	1	Látkové množství	6
Vzorec	2	Aktivita a Koncentrace roztoku	6
Vzorec	3	Konstanta sondy	6
Vzorec	4	Molární konduktivita	7
Vzorec	5	Výpočet konduktivity	7
Vzorec	6	Rychlost komunikace po sériové lince	21
Vzorec	7	Výstupní frekvence	35
Vzorec	8	Zesílení AMP	36
Vzorec	9	Zesílení AMP	36
Vzorec	10	Zesílení AMP	36
Vzorec	11	Odpor sondy	36
Vzorec	12	Odpor sondy	36
Vzorec	13	Zesílení napětí	41
Vzorec	14	Dělič napětí	41

Seznam symbolů, veličin a zkratk

DPS	deska plošných spojů
pH	označení potenciálu vodíků
EC	elektrická vodivost
Elektrolyt	roztok
Log	logaritmus
S/cm	jednotka elektrické vodivosti Siemens/centimetr
g	jednotka hmotnosti, gram
l	jednotka objemu, litr
H ₂ O	chemické značení vody
KCl	chemické značení Chloridu draselného
Atmel	firma vyrábějící mikroprocesory
Flash paměť	elektronicky programovatelná paměť
USBtoUART	rozhraní pro propojení USB kanálu a UART kanálu
USB	univerzální sériové rozhraní používané v počítačích
UART	univerzální sériové rozhraní
ADC	analogově-digitální převodník
FW	firmware
SW	software
MHz	jednotka frekvence
LC filtr	filtr obsahující tlumivku a kondenzátor
ESD	ochrana proti elektro-statickému výboji
Baud	jednotka znakové rychlosti
Null znak	nulový znak
Datasheet	technická dokumentace součástky/zařízení
Int	datový typ
C#	programovací jazyk
Label	pole pro výpis hodnoty
Byte	datová jednotka obsahující 1 bit
Bit	nejmenší datová jednotka, nabývá hodnot 0 nebo 1 (ano či ne)

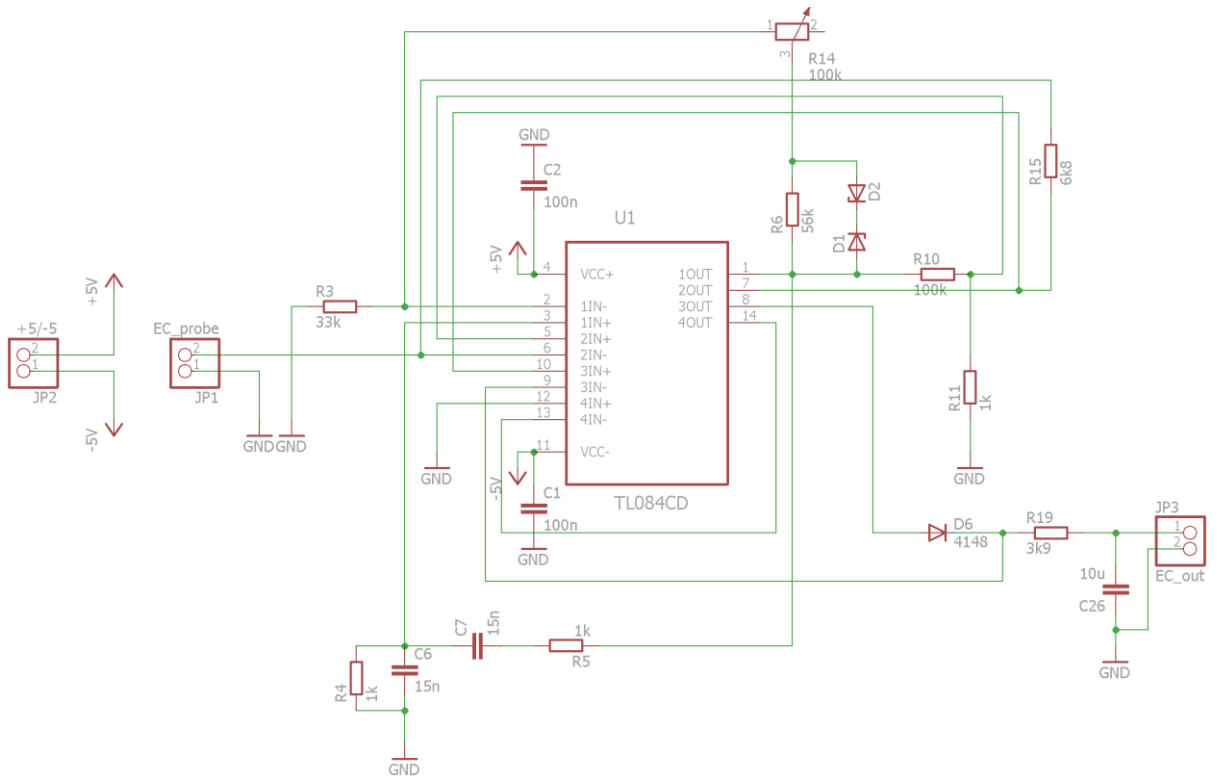
SerialPort	port na kterém probíhá komunikace sériově neboli „data jdou za sebou“
comboBox	roletové menu
string	řetězec
buffer	vyrovnávací paměť
JFET	unipolární zesilovač
Rail to Rail	výstupní napětí může dosahovat téměř napájecího napětí

Seznam příloh

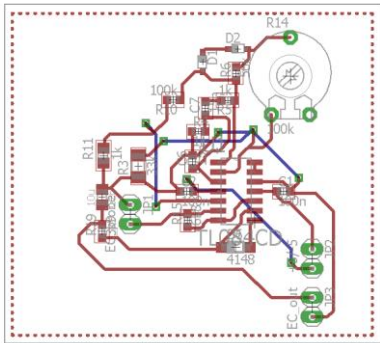
Přílohy jsou obsažené na CD disku přiloženém k práci.

1. EC schéma
2. EC deska
3. pH schéma
4. pH deska
5. Power deska
6. Power schéma
7. FW
8. SW

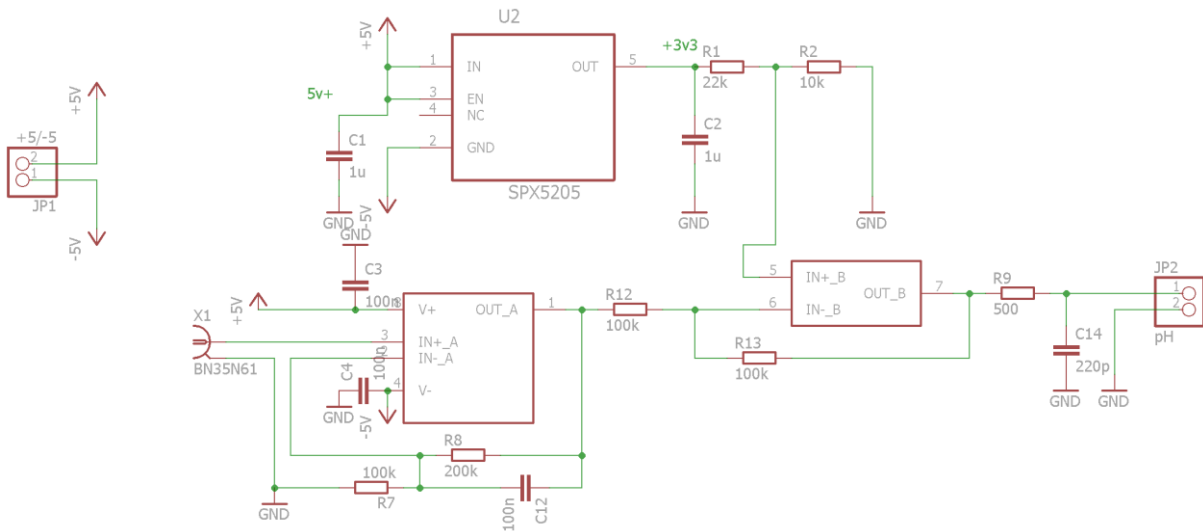
- 1.



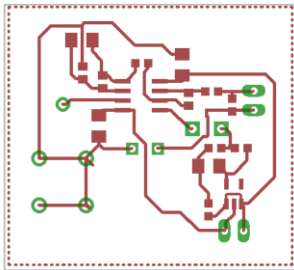
2.



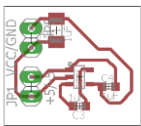
3.



4.



5.



6.

