

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE  
TECHNICKÁ FAKULTA

KATEDRA ELEKTROTECHNIKY A AUTOMATIZACE



## DIPLOMOVÁ PRÁCE

Závlahový systém pro zimní zahradu

**Autor:** Michal Bůžek

**Vedoucí práce:** doc. Ing. Stanislava Papežová, CSc.

Praha, 2014

# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra elektrotechniky a automatizace

Technická fakulta

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Bůžek Michal

Informační a řídicí technika v agropotravinářském komplexu

Název práce

**Závlahový systém pro zimní zahradu**

Anglický název

**Watering system for winter garden**

---

### Cíle práce

Cílem práce je návrh a realizace zařízení pro regulaci vlhkosti půdy pro účely pěstování rostlin. Zařízení bude umožňovat udržování vlhkosti půdy v závislosti na čase podle průběhu zadaného uživatelem.

### Metodika

Stanovte technické požadavky navrhovaného systému. Zvolte vhodný způsob vyhodnocování půdní vlhkosti a způsob jejího měření. Proveďte výběr vhodného mikroprocesoru a akčního členu. Zařízení realizujte a vybavte jej příslušným programovým vybavením pro daný účel. Ověřte správnou funkci celé soustavy a vyhodnoťte výsledky.

### Osnova práce

1. Úvod
2. Cíl práce
3. Rozbor požadavků na regulaci půdní vlhkosti
4. Senzory pro indikaci vlhkosti půdy, volba vhodného senzoru s ohledem na požadavky zadání
5. Výběr řídicí jednotky zařízení a komunikačního rozhraní pro zadávání křivky vlhkosti
6. Konstrukční návrh a realizace zařízení
7. Softwarové vybavení systému
8. Ověření funkčnosti zařízení a její vyhodnocení
9. Závěr
10. Seznam odborné literatury

## **Rozsah textové části**

40 - 50 stran

## **Klíčová slova**

vlhkost půdy, závlaha, mikrokontrolér, řízení

---

## **Doporučené zdroje informací**

Kreidl Marcel: Senzory, ČVUT v Praze, 1992

John Davies: MSP430 Microcontroller Basics, 1st Edition, Newnes, 2008

Císlerová, M.: Inženýrská hydroopedologie, skriptum ČVUT 2001

Kutílek, M., Kuráž, V., Císlerová, M.: Hydroopedologie, skriptum ČVUT, 1994

Fexa, J., Široký, K.: Měření vlhkosti, Praha, Nakladatelství technické literatury, 1983, 262 s.

---

## **Vedoucí práce**

Papežová Stanislava, doc. Ing., CSc.

## **Termín zadání**

listopad 2012

## **Termín odevzdání**

duben 2014

---

Elektronicky schváleno dne 15.1.2014

**prof. Ing. Jaromír Volf, DrSc.**

Vedoucí katedry

---

Elektronicky schváleno dne 24.2.2014

**prof. Ing. Vladimír Jurča, CSc.**

Děkan fakulty

## Prohlášení

Prohlašuji, že jsem práci na téma „Závlahový systém pro zimní zahradu“ vypracoval samostatně a použil jsem podklady uvedené v seznamu literatury.

Nemám žádný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 6. 4. 2014

.....

## **Poděkování**

Děkuji především za odborné konzultace a velmi vstřícný přístup vedoucí mé diplomové práce docentce Ing. Stanislavě Papežové CSc.

## Abstract

The goal of this thesis is to design an irrigation device that can control soil moisture by entering its control curve. The first part of this paper is concerned with hardware development of the irrigation device and selection of its elements according to the requirements. The following part aims on development of control firmware for the MSP430 microcontroller that controls the irrigation device. The third part is dedicated to designing a control software for personal computer to which the device can be connected. A functional prototype of low-cost irrigation device has been created. This prototype can irrigate soil according to the soil moisture curve entered by a computer programme.

**keywords:** microcontroller, soil moisture, automatic irrigation

## Abstrakt

Práce se věnuje tvorbě závlahového zařízení, které umožňuje řídit závlahu zadáním řídicí křivky půdní vlhkosti. První část práce se zabývá vývojem hardwaru závlahového zařízení a výběru jeho prvků podle zadaných požadavků. Následující část se zabývá vývojem řídicího firmware pro mikroprocesor MSP430, který řídí závlahové zařízení. Třetí část se zabývá návrhem ovládacího softwaru pro osobní počítač, ke kterému je možné zařízení připojit. Podařilo se vyrobit funkční prototyp levného zařízení, které dokáže zavlažovat půdu podle křivky zadané v počítači.

**Klíčová slova:** půdní vlhkost, automatické zavlažování, mikroprocesor

## Obsah

<b>1 Úvod</b>	<b>1</b>
1.1 Motivace . . . . .	1
1.2 Cíl a metodika práce . . . . .	1
1.3 Struktura práce . . . . .	2
<b>2 Teorie</b>	<b>3</b>
2.1 Požadavky rostlin . . . . .	3
2.2 Měření půdní vlhkosti . . . . .	5
2.2.1 Gravimetrická metoda . . . . .	6
2.2.2 Vodivostní metoda . . . . .	6
2.2.3 Metoda TDR . . . . .	6
2.2.4 Metoda TDT . . . . .	7
2.2.5 Metoda ADR . . . . .	7
2.2.6 Kapacitní metody . . . . .	7
2.2.7 Další metody . . . . .	8
2.3 Požadavky na systém . . . . .	8
<b>3 Hardware</b>	<b>9</b>
3.1 Výběr mikrokontroléru . . . . .	9
3.2 Výběr senzoru půdní vlhkosti . . . . .	10
3.2.1 „Ebay“ China sensor . . . . .	10
3.2.2 Watermark 6440 . . . . .	10
3.2.3 Vegetronix VH400 . . . . .	11
3.2.4 VIRRIB . . . . .	12
3.2.5 Vodivostní senzor vlastní konstrukce . . . . .	12
3.2.6 Kapacitní senzor vlastní konstrukce . . . . .	13

---

3.3	Výběr akčního členu . . . . .	14
3.3.1	Čerpadlo . . . . .	14
3.3.2	Elmag. ventil . . . . .	15
3.4	Návrh řídicí jednotky . . . . .	15
3.4.1	Schéma řídicí jednotky . . . . .	15
3.4.2	Napájení zařízení . . . . .	16
3.4.3	Vstupy pro senzory . . . . .	18
3.4.4	Měřící obvod . . . . .	18
3.4.5	Výstupy pro akční členy . . . . .	19
3.4.6	Komunikace s PC . . . . .	19
3.4.7	Deska plošného spoje . . . . .	19
3.4.8	Hotový prototyp hardware . . . . .	20
<b>4</b>	<b>Firmware mikroprocesoru</b>	<b>21</b>
4.1	Hlavní smyčka programu . . . . .	21
4.2	Přerušování časovače . . . . .	22
4.3	Přerušování GPIO . . . . .	24
4.4	Přerušování UART . . . . .	24
4.5	Zápis do flash paměti . . . . .	25
4.6	Procházení tabulky v paměti . . . . .	27
4.7	Regulace půdní vlhkosti . . . . .	28
4.8	Detekce napájecího napětí . . . . .	28



---

<b>5 Software pro tvorbu křivky</b>	<b>30</b>
5.1 Výběr vývojového prostředí . . . . .	30
5.2 Návrh uživatelského rozhraní . . . . .	30
5.2.1 Požadavky . . . . .	30
5.2.2 Vzhled a rozmístění prvků . . . . .	30
5.2.3 Hlavní okno . . . . .	30
5.2.4 Okno komunikace se zařízením . . . . .	31
5.2.5 Práce se soubory . . . . .	32
5.3 Realizace . . . . .	32
5.3.1 Objektový návrh . . . . .	32
5.3.2 Datové operace . . . . .	32
5.3.3 Zobrazení dat . . . . .	34
5.3.4 Komunikace se zařízením . . . . .	35
<b>6 Ověření funkčnosti zařízení</b>	<b>37</b>
6.1 Měření vlhkosti půdy . . . . .	37
6.1.1 Vliv umístění senzoru . . . . .	37
6.1.2 Kontrola kvality měření a regulace . . . . .	37
6.1.3 Test stability měřených veličin . . . . .	38
6.2 Regulace podle křivky . . . . .	39
6.3 Vyhodnocení . . . . .	40
<b>7 Závěr</b>	<b>41</b>
<b>Reference</b>	<b>42</b>

# 1 Úvod

*„Žádný zázrak slunce nevyrovná se zázraku požehnaného deště. Běž, zkalená vodičko, stružkami země, napoj a zkypři žíznivou hmotu, jež nás vězní. Všichni jsme vydechli, já, hlína, my všichni, je nám dobře.“*

Karel Čapek: Zahradníkův rok

Voda je základním předpokladem pro vznik života. Rostliny jí potřebují k životu stejně tak jako člověk. Přírozené srážky často nestačí na pokrytí potřeb květin a proto je potřeba dodat květinám chybějící vláhu v době sucha. Proces umělého zavlažování je asi tak starý jako samotné zemědělství a již tisíce let před naším letopočtem zemědělci vymýšleli, jak tento proces mechanizovat. Vznikaly různé zavlažovací kanály, pumpy a čerpadla poháněná zvířaty a podobně. Dnes je běžnou součástí zemědělství i zcela automaticky řízená závlaha, kdy počítač řídí přesnou závlahovou dávku pro systém postřikovačů, tak aby se spotřebovalo jen tolik vody, kolik rostliny skutečně potřebují.

## 1.1 Motivace

Na Technickou fakultu České zemědělské univerzity jsem se přihlásil proto, že mě lákalo spojení techniky s přírodou. Chtěl jsem proniknout do tajů automatického řízení procesů, které jsme zvyklí dělat ručně. Zalévání je typický příklad takového procesu, který se dá automatizovat. Na trhu sice existují komerčně prodávané závlahové automaty. Jsou ale drahé (v řádech tisíců korun jen za řídicí jednotku) a především, naprostá většina z nich vůbec neřeší vlhkost půdy. Zalévání je naprogramováno časově, případně je možnost blokovat zalévání při zapojení dešťového senzoru. Velmi drahé systémy pro velké zemědělské podniky mohou být vybaveny i senzory vlhkosti půdy, avšak tam je cena ještě vyšší. Na trhu zjevně chybí systém pro drobné pěstitele, zahrádkáře, či do domácnosti, který by řešil množství závlahy podle půdní vlhkosti s možností nastavení pro různé druhy rostlin za adekvátně nízkou cenu. Rozhodl jsem se proto takové zařízení vytvořit.

## 1.2 Cíl a metodika práce

Tato práce se zabývá návrhem levného závlahového systému pro použití v domácnosti, nebo na zahradě. Na začátku se věnuje rozboru požadavků rostlin a z nich vyplývajících požadavků na závlahový systém.

Poté se věnuje samotnému návrhu zařízení od výběru komponent, přes vývoj hardwaru a firmwaru až po hotový prototyp. Následuje část věnovaná obslužnému software do počítače a končí ověřením funkčnosti zařízení a vyhodnocení úspěšnosti práce.

Cílem práce je vytvořit funkční prototyp závlahového zařízení schopný řídit závlahovou dávku podle křivky půdní vlhkosti v čase. Zařízení musí být připojitelné k osobnímu počítači. Nastavení zařízení a vkládání řídicí křivky bude probíhat pomocí počítače.

### 1.3 Struktura práce

2. kapitola se zabývá teoretickým základem a požadavky na vyvíjené zařízení.

Ve 3. kapitole se z počátku věnují výběrem vhodného senzoru půdní vlhkosti a akčního členu pro zavlažování s přihlédnutím na zdroj vody. Poté se věnují návrhu a realizaci hardwaru řídicí jednotky spolu s popisem a řešením problémů, které se vyskytly v průběhu návrhu a realizace.

4. kapitola popisuje řídicí firmware mikroprocesoru a také softwarová řešení víceméně hardwarových problémů, které se při realizaci vyskytly.

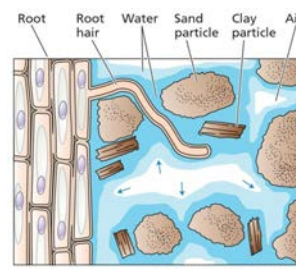
5. kapitola se věnuje návrhu programu pro osobní počítač. Pomocí tohoto programu je možné nastavit regulační konstanty zařízení, nastavit hodiny v zařízení a především vytvořit regulační křivku půdní vlhkosti a tu následně uložit do zařízení.

## 2 Teorie

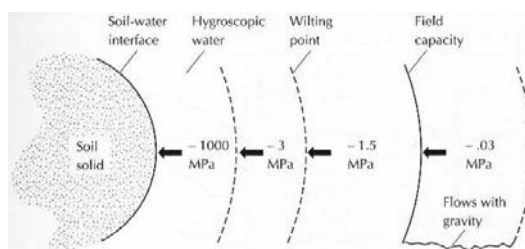
### 2.1 Požadavky rostlin

Vlhkost půdy je jednou z nejdůležitějších charakteristik prostředí ovlivňující vývoj rostliny. Vedle působení klimatických prvků (srážky a výpar) je hydrický režim významně ovlivněn půdními vlastnostmi. Naopak půdní voda má velký význam pro vývoj půdy a utváření půdních vlastností. Obsah půdní vody je v podmínkách zemědělské produkce rovněž velmi ovlivňován způsoby zpracování půdy; příznivý vliv na udržení vody v půdě mají zejména technologie omezující orbu. Obsah vody v půdě je nejčastěji uváděn v objemových procentech. Takto zjištěné množství vody ale není v celém svém rozsahu k dispozici i rostlinám. Přístupnost vody pro rostliny je omezena půdním hydrolimitem označeným jako bod vadnutí (Wilting point). Určuje vlhkost půdy, kdy rostliny již nejsou schopny využít půdní vodu, absorpce vody kořenovým systémem je podstatně nižší než intenzita transpirace a rostliny vadnou. Protože vadnutí rostlin probíhá v dosti širokém intervalu vlhkosti, udává se jako mezní limit spodní mez tohoto intervalu [10]. Pod hodnotu bodu vadnutí se půda prostřednictvím evapotranspirace rostlin již prakticky nevysouší. Rostlinám k dispozici tedy není půdní voda určená pouze aktuální vlhkostí půdy, ale voda určená jako rozdíl aktuální vlhkosti půdy a bodu vadnutí. Minimální množství volné vody v půdě je tzv. hygroskopická voda v půdě, neboli kapilární/adsorpční voda.

Druhým důležitým půdním hydrolimitem je retenční vodní kapacita (někdy označována jako polní kapacita z anglického Field capacity). Specifikuje maximální množství vody, které je půda schopna pojmout [10]. Při této vlhkosti půdy jsou v půdě naplněny všechny kapilární póry. Při dalším zvlhčení půdy již voda může pronikat pouze do nekapilárních pórů; ty jsou schopny ji udržet jen krátkodobě a tato voda postupně protéká půdním profilem do spodních vrstev až případně do úrovně hladiny podzemní vody (v květináči vyteče do podtácku). Proto v půdě téměř vždy určitá část půdních pórů zůstává vyplněna vzduchem. Toto je ale dobře, protože většina rostlin ke svému vývoji potřebuje provzdušněnou půdu, proto se také provádí orba a podobné činnosti. Optimální provzdušněnost je kolem 30 % celkové pórovitosti půdy [10]. Maximální množství vody v půdě využitelné rostlinami je dáno rozdílem půdních vlhkostí při retenční vodní kapacitě a bodu vadnutí. Tento rozdíl se nazývá využitelná vodní kapacita půdy (na obr. 3 vyznačena světle šedivou barvou pod pojmem „water available to plants“). Může se rovněž uvádět v objemových procentech, avšak velmi často se udává v mm vodního sloupce vztaženého k určité hloubce půdního profilu (nejčastěji 1 m).

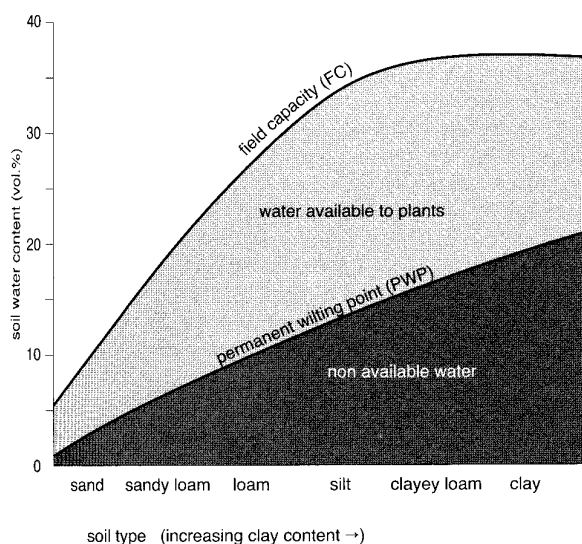


Obrázek 1: Struktura zeminy [17]



Obrázek 2: Půdní hydrolimity

Půdní hydrolimity jsou významně ovlivněny fyzikálními a chemickými vlastnostmi půdy. Velký význam má zvláště půdní druh. Písčité půdy (tzn. půdy lehké), obsahující větší půdní částice, mají oba zmíněné hydrolimity (bod vadnutí a retenční vodní kapacitu půdy) nízké. Naopak jílovité půdy (tzn. půdy těžké), obsahující ve větší míře menší půdní částice, mají oba hydrolimity vysoké (viz. obr. 3). U jílovitých půd je problémem právě vysoká hodnota retenční vodní kapacity, respektive nízká retenční vzdušná kapacita, která je dána rozdílem pórovitosti a retenční vodní kapacity. Tato skutečnost způsobuje, že během roku mohou nastat období, kdy je většina půdních pórů vyplněna vodou a v půdě je nedostatek vzduchu. U písčitých půd k tomuto nedochází. Nízké hydrolimity u písčitých půd by samy o sobě nebyly problémem; ten vzniká díky tomu, že rozdíl obou hydrolimitů charakterizující využitelnou vodní kapacitu je u písčitých půd malý. Z pohledu vývoje rostlin jsou ideálními půdami půdy hlinité až hlinitopísčité (tzn. půdy středně těžké), které disponují vysokou hodnotou využitelné půdní kapacity a rovněž jejich retenční vodní kapacita je vůči pórovitosti dostatečně nízká, aby ani ve vlhčích obdobích roku nedocházelo k přemokření půdy [13].

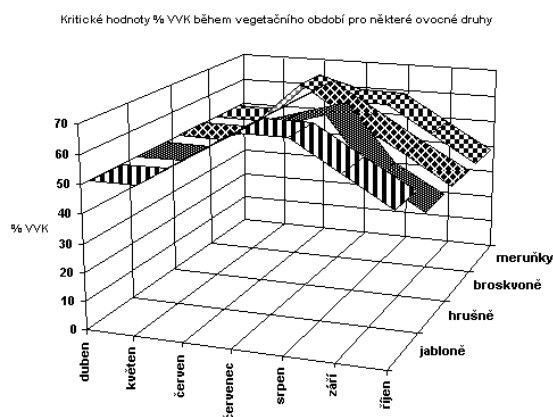


Obrázek 3: Závislost množství vody přístupné rostlinám na typu zeminy a vlhkosti půdy[9]

Při analýze půdních vlhkostí je důležité určit půdní hydrolimity sledované půdy. Například relativní vlhkost 20 % může u písčitých půd představovat stav vysokého množství vody v půdě, naopak u půd jílovitých půd z pohledu rostlin zcela suchou. Jak bylo řečeno, písčité půdy disponují nízkými hodnotami využitelné vodní kapacity (VVK). Tato skutečnost je příčinou, že půda je schopna zadržet velmi omezené množství vody. V případě bezsrážkové epizody spojené s vyšším výparem stačí nedlouhá doba k tomu, aby se půda stala z pohledu potřeby rostlin suchou. Naopak při vydatnějších deštích se půdní profil písčité půdy rychle zaplní a přebytečná voda rychle odtéká, aniž by ji rostliny mohli využít [13].

Pro úspěšné pěstování rostlin je mimo jiné potřeba zajistit správnou vlhkost půdy. Různé druhy rostlin se ve svých nárocích na půdní vlhkost značně liší. Optimální úroveň půdní vlhkosti nezávisí jen na složení půdy, její struktuře a pórovitosti, ale i na vegetačním období teplotě vzduchu a půdy vlhkosti vzduchu a dalších faktorech.

Většina rostlin má ráda nakypřenou půdu, důkladnou zálivku načasovanou v rámci dne na ráno, či večer (večerní zálivka nemusí být tak důkladná, protože odpařování během noci je nižší a při vysoké



Obrázek 4: Průběh kritických hodnot VVK během vegetačního období pro různé plodiny

vlhkosti se zvyšuje riziko plísní a dalších chorob) a také je dobré provádět závlivku vícefázově. To vyplývá ze sací schopnosti půdy, která není schopná pojmout najednou velké množství vody v krátkém čase, zkrátka potřebuje čas na vsáknutí, po vsáknutí je možné pokračovat v zalévání. Pokud bychom provedli velkou závlivku najednou hrozilo by odplavování úrodné vrstvy půdy (eroze).

Některé rostliny mají rády sucho, jiné mokro, většinou rostlin pak trvalá vlhkost nedělá dobře, mají raději biorytmus závlahových cyklů, kdy mezi závlahami stihne půda proschnout. To podporuje provzdušnění kořenů a podmínky vhodné pro rozvoj mikroorganismů. V tabulce 1 je srovnání požadavků různých druhů běžně pěstovaných rostlin na velikost závlivky.

Skupina	Druh	Závlahová dávka v litrech/m <sup>2</sup>	Správná hloubka provlhčení v m
tráva		20 - 30	0,15 - 0,2
ovocné dřeviny	meruňky, broskvoně, jabloně, třešně	30 - 50	0,4 - 0,6
bobuloviny	rybíz, angrešt, maliny, jahody	15 - 30	0,2 - 0,4
kořenová zelenina	mrkev, celer, petržel	25 - 50	0,2 - 0,4
cibulová zelenina	cibule, česnek, pór	10 - 25	0,1 - 0,2
lusková zelenina	hrách, fazole	20 - 40	0,1 - 0,3
plodová zelenina	rajčata, okurky, paprika	20 - 40	0,1 - 0,3
košťálová zelenina	květák, zelí, kapusta	15 - 40	0,1 - 0,3

Tabulka 1: Požadavky různých druhů rostlin na závlivku[18]

## 2.2 Měření půdní vlhkosti

Po vysvětlení, jak půda zadržuje vlhkost vzniká otázka, jak se půdní vlhkost dá měřit. Jsou dva způsoby vyjádření půdní vlhkosti:

- **Objemová vlhkost** - podíl objemu vody k celkovému objemu půdy
- **Hmotnostní vlhkost** - podíl hmotnosti vody k hmotnosti tuhé fáze půdy (někdy také vlhkost relativní)

Protože chceme měřit půdní vlhkost proto, abychom zjistili, jestli mají rostliny dostatek vody, chtěli bychom zjistit množství vody dostupné kořenům. Ideální by bylo pro měření použít umělý měřicí kořen. Velmi přesný způsob jak toho dosáhnout je použití půdního tenzometru. Ten nám nedá za výsledek hodnotu absolutní vlhkosti, ale sací tlak potřebný pro získání vody ze substrátu. Avšak tyto tenzometry jsou drahá zařízení, která vyžadují vědecký přístup, proto se běžně používají jiné metody.

### 2.2.1 Gravimetrická metoda

Jednou z nejpřesnějších metod určování vlhkosti půdy je metoda gravimetrická. Jedná se o přímou metodu měření. Princip spočívá ve zvážení vzorku půdy o zjišťované vlhkosti, vysušení vzorku při teplotě 105 až 110 °C na konstantní hmotnost suché půdy. Podíl těchto dvou hmotností je hodnota hmotnostní vlhkosti. Vzhledem ke své přesnosti je metoda často používána ke kalibraci senzorů založených na nepřímých metodách měření. Nevýhodou této metody je pracnost, časová náročnost a není možné ji použít k průběžnému vyhodnocování stavu vlhkosti půdy.

### 2.2.2 Vodivostní metoda

S rostoucí vlhkostí roste i vodivost půdy. Toho využívá vodivostní metoda měření. Závislost měrného odporu půdy na vlhkosti je nelineární, naměřená hodnota dále závisí na chemickém složení (s rostoucím obsahem solí odpor obvykle klesá), na okolní teplotě a na struktuře půdy. Proto je velmi obtížné tato čidla nakalibrovat. Ve vlhké půdě elektrody z nevhodného materiálu korodují působením elektrolýzy, čímž se zvyšují přechodové odpory a snižuje se životnost sondy. Vhodný materiál pro výrobu elektrody musí být chemicky stálý (nerez, zlato, platina), což ve výsledku může ovlivnit cenu senzoru. Korozi lze omezit použitím střídavého měřicího proudu a měřením pouze v nutných časových intervalech. Působení solí na měřené hodnoty se dá za cenu zvýšení reakční doby omezit umístěním elektrod do svého materiálu (typicky sádra), který brání solím dostat se až k elektrodám. Tato metoda je velmi rozšířená, protože senzory jsou levné a i bez kalibrace její vypovídací hodnota dostatečná pro účely automatického zavlažování.

### 2.2.3 Metoda TDR

Měření metodou TDR (Time Domain Reflectometry) byla vyvinuta pro lokalizaci poškození kabelu. Měří se doba odrazu elektromagnetického pulzu a z ní je možné vypočítat vzdálenost poškozeného místa v kabelu, dá se tak zjistit i místo, kde do kabelu vnikla voda díky změně dielektrických vlastností vodiče, což ovlivňuje rychlost šíření pulzu. Změnu dielektrických vlastností půdy je možné použitím metody TDR měřit také. TDR senzor půdní vlhkosti se obvykle skládá ze dvou rovnoběžných hrotů, které se chovají jako vyvážené vedení a dielektrikum tvoří zemina. Jeden hrot vysílá sérii přesně časovaných vysokofrekvenčních pulzů a měří dobu jejich návratu po odrazu od druhého hrotu po průchodu půdou. Takto získaný signál musí elektronika senzoru zpracovat. Vzhledem k faktu, že tyto jevy se odehrávají v řádech Mhz až Ghz a elektronika musí v této rychlosti pracovat a vypočítávat výsledek, je cena senzoru vysoká. Výhodou této metody je vysoká rychlost odezvy a vysoká přesnost bez nutné kalibrace, ta je nutná pouze pro půdu s vysokou koncentrací minerálních látek (salinitou).

### 2.2.4 Metoda TDT

Měření metodou TDT (Time Domain Transmissometry) je založené na měření fázového posunu sinusového signálu po průchodu přenosovým vedením umístěným v substrátu. Fázový posun závisí na frekvenci, rychlosti šíření v dielektriku a délce přenosového vedení. Při neměnné frekvenci a délce přenosového vedení může být vlhkost půdy určena z fázového posunu. Jedná se vlastně o modifikaci metody TDR tak, aby zařízení na měření nemuselo mít takový výpočetní výkon. Čidlo je obvykle tvořeno dvěma kovovými soustřednými prstenci (např. senzor VIRRIB zmíněný v kapitole 3.2.4), takže elektronika vyhodnocující fázový posun může být připojena na začátku a konci vlnovodů. Výhodou metody je vysoká přesnost bez nutnosti kalibrace (lze kalibrovat pro jiné materiály, než je půda, případně pro měření v nestandardní půdě), výstup z vyhodnocovací elektroniky je linearizovaný a snadno dále zpracovatelný, měří v poměrně velkém objemu okolní půdy. Tyto senzory nejsou vyloženě drahé a podávají velmi dobré a stabilní výsledky.

### 2.2.5 Metoda ADR

Metodou ADR se měří elektrická impedance. Vysokofrekvenční elektromagnetické vlny sinusového průběhu se šíří přenosovým vedením (transmission line). Tyto vlny vstupují do půdy s odlišnou impedancí od samotného přenosového vedení a část energie se odráží zpět k vysílači. Odražená vlna interferuje s původní vlnou čímž vytváří stojaté vlnění podél přenosového vedení. Změna amplitudy vlnění pak určuje impedanci. Impedance se skládá ze dvou složek - elektrické vodivosti a dielektrické konstanty. Volbou vhodné frekvence lze potlačit vliv vodivosti. Výsledné změny v amplitudě pak závisí jen na dielektrických vlastnostech a ty se mění s vlhkostí půdy. Výhodou této metody je poměrně přesné měření i bez nutné kalibrace. Podává dobré výsledky měření i v půdách s vysokou salinitou a výsledek měření není nezávislý na teplotě. Výsledek měření ale může být zkreslen nehomogenitou půdy, kdy se v měřené půdě vyskytují větší vzduchové bubliny, kameny apod..

### 2.2.6 Kapacitní metody

Relativní permitivita půdy se mění v závislosti na obsahu vody v půdě. Změna relativní permitivity se v praxi vyhodnocuje jako změna kapacity senzoru (kondenzátoru), jehož dielektrikum tvoří půda. Desky kondenzátoru jsou elektricky odizolované od půdy, tudíž nehrozí jejich koroze. Změna kapacity kondenzátoru může být vyhodnocována buď metodou **FD** (Frequency Domain Capacitance), nebo metodou **FDR** (Frequency Domain Reflectometry). Při měření metodou FD určuje senzor (kondenzátor) přímo časovou konstantu oscilátoru a tím mění i jeho frekvenci, kterou měříme. Při měření metodou FDR metody je kondenzátor součástí rezonančního obvodu, který je napájen z oscilátoru s proměnlivou frekvencí a stanovení kapacity probíhá hledáním rezonanční frekvence obvodu. Závislost kapacity senzoru na vlhkosti není lineární. Výsledky měření také ovlivňuje teplota, chemické složení a pórovitost půdy. Kalibrace pro různé druhy půdy je pro tento typ senzorů nutná. Kapacitní měření má výbornou reakční dobu a vysoký rozsah měření s vysokým rozlišením. Cena senzorů je nižší než u senzorů na principu TDR. Životnost senzorů je vyšší, než u vodivostní metody. Nevýhodou zůstává ovlivnění měření při nehomogenitě půdy (permitivita vzduchu je výrazně nižší než permitivita vlhké půdy, což při výskytu vzduchových bublin v zemině hraje významnou roli).



### 2.2.7 Další metody

Další zajímavou metodou měření vlhkosti půdy je GPR (Ground Penetrating Radar). Ta dokáže scanovat vlhkost půdy do hloubky bez nutnosti dotyku senzoru se zemí. Při měření vytváří 2D hloubkové profily vlhkosti a kombinací takto navzorkovaných profilů lze dosáhnout 3D mapy vlhkosti půdy [4]. Kvalita měření se dá srovnat s TDR senzory. Toto je vhodné pro mobilní měření, kdy se vozíčkem s GPR senzorem dá změřit vlhkostní profil pole.

Jiné metody jako zpomalování neutronů (Neutron Moderation), rentgenová tomografie nebo NMR (Nuclear Magnetic Resonance) jsou příliš komplikované a drahé pro použití v levném závlahovém systému.

## 2.3 Požadavky na systém

V současné době existují řídicí jednotky pro řízení množství závlahy od různých výrobců a je jich mnoho. Oblíbené jsou například Bosh abc-4000, či Gardena 1060 [18], žádný z těchto systémů ale nepodporuje řízení půdní vlhkosti podle její křivky. Lepší řídicí jednotky dokáží blokovat časově nastavený zavlažovací proces podle hodnoty dešťového senzoru, či senzoru půdní vlhkosti, popřípadě dokáží podle senzoru půdní vlhkosti regulovat jednu konkrétní nastavenou hodnotu.

Navrhovaný systém by měl zvládat, možnost navolení křivky půdní vlhkosti v čase, kterou pak bude systém udržovat. Tím by měla být zajištěna flexibilita při výběru druhu pěstovaných rostlin. Křivka vlhkosti pak bude odpovídat profilu potřeb vybrané rostliny. Měl by být levný, tak aby byl dostupný každému a jednoduchý na obsluhu a nastavení. Systém by měl být co nejvíce přizpůsobitelný potřebám různých nasazení. Možnost připojení různých druhů čidel a akčních členů, či provoz z baterie i z rozvodné sítě je více než vítána.



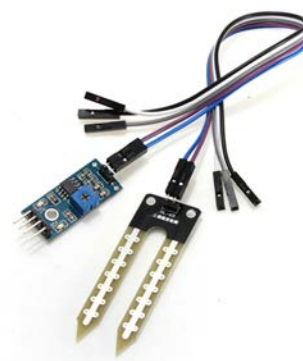
MSP430 Derivative	Flash (bytes)	RAM (bytes)	16-Bit Timers	Common Peripherals	ADC	Additional Features	Packages
F550x, F5510	8K-32K	4K(+2K)	4	<ul style="list-style-type: none"> <li>• USB</li> <li>• WDT</li> <li>• RTC</li> <li>• DMA (3-6)</li> <li>• MPY32</li> <li>• Comp_B</li> <li>• UART</li> <li>• SPI</li> <li>• I2C</li> <li>• PMM (BOR, SVS, SVM, LDO)</li> </ul>	10-bit SAR		<ul style="list-style-type: none"> <li>• 48RGZ</li> <li>• 48PT</li> <li>• 64RGC</li> </ul>
F551x, F552x	32K-256K	4K-8K(+2K)			12-bit SAR	<ul style="list-style-type: none"> <li>• 80PN</li> <li>• 64RGC</li> <li>• 80ZQE</li> <li>• 64YFD (chip-scale DSBGA)</li> </ul>	
F563x, F663x	128K-256K	16K(+2K)			<ul style="list-style-type: none"> <li>• EDI</li> <li>• DAC12</li> <li>• Backup battery switch</li> <li>• LCD (on F663x and F665x)</li> </ul>	<ul style="list-style-type: none"> <li>• 100PZ</li> <li>• 113ZQW</li> </ul>	
F565x, F665x	384K-512K	32K-64K(+2K)					

Tabulka 2: Mikroprocesory MSP430 vybavené USB [11]

## 3.2 Výběr senzoru půdní vlhkosti

### 3.2.1 „Ebay" China sensor

Jako první senzor, kterého je plný internet, jsem zahlédl čínský senzor z obrázku 6. Prodává se hojně na aukčním serveru ebay.com a čínských e-shopech jako je aliexpress.com nebo dealextreme.com za cenu do **2 USD**. Senzor z obrázku je nejčastější ale objevují se i jiné, avšak velmi podobné. Výrobce ani typ produktu prodejci neuvádějí. Objednal jsem ho víceméně ze zvědavosti, na vyzkoušení. Za cenu 30 Kč proč ne? Bylo mi jasné už z fotografie a podle ceny, že to s kvalitou nebude moc valné.



Obrázek 6: Čínský senzor z Ebaye

Nemýlil jsem se, detekční hroty na destičce plošných spojů začínají oxidovat již po pár týdnech používání. Na první pohled se jejich povrch jeví jako pouze pocínovaný. Životnosti elektrod navíc velmi neprospívá ani měření stejnosměrným proudem, kdy probíhá v substrátu elektrolýza jedním směrem. Z principu elektrolýzy vyplývá, že na jedné elektrodě probíhá redukce a na druhé oxidace. Doufal jsem, že obvod na přiložené destičce zařizuje měření střídavým proudem, avšak je to pouze komparátor LM393, který zajišťuje digitální výstup, který mění stav 1/0 podle nastavené hodnoty potenciometrem.

Z provedených měření mě ale překvapil, že alespoň trochu funguje - i když nepřesně. Otázkou je jak dlouho bude. Jisté je, že v průběhu používání se budou měřené hodnoty měnit vzhledem k oxidaci kontaktů.

### 3.2.2 Watermark 6440

Nevýhodou senzorů na principu měření vodivosti, stejnosměrným i střídavým proudem je, že měřené hodnoty jsou ovlivněné množstvím soli v substrátu. Proto se používají takzvané „gypsum blocky" (česky by se dalo přeložit jako sádrové bločky). Jedná se o váleček ze savého materiálu (např. sádra)



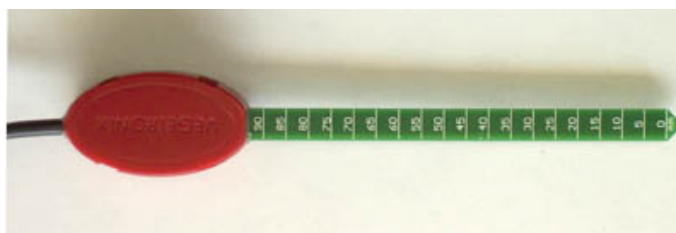
Obrázek 7: Watermark 6440 (Gypsum block)

uvnitř kterého jsou umístěny nerezové elektrody, které se využívají pro měření vodivosti. Tím, že elektrody jsou uvnitř tohoto materiálu se částečně řeší problém se slaností substrátu. Blok materiálu kolem elektrod tvoří jakousi bariéru pro sůl, která se tak nedostane až k elektrodám. Problém ale je velmi dlouhá reakční doba, kdy musí celý blok nasáknout vlhkost substrátu a to může trvat velmi dlouho.

Komerčně prodáváný Watermark 6440 za **1 749 Kč** [2] mi připadá příliš drahý. V kontrastu s „home-made“ [1] verzí vyrobené z páru hřebíků (doporučoval bych nerezové) a sádky se zdá přímo předražný. Já však tento typ senzoru vyloučil pro jeho dlouhou reakční dobu. Pokud by však dlouhá reakční doba nebyla na obtíž, nic nebrání připojení tohoto druhu senzoru k prvnímu vstupu řídicí jednotky, který převádí vodivost na měřenou frekvenci.

### 3.2.3 Vegetronix VH400

<<http://vegetronix.com/Products/VH400/>>



Obrázek 8: Vegetronix VH400

Z komerčních produktů na mě nejlépe působil senzor půdní vlhkosti VH400 od firmy Vegetronix. Podle reakcí zákazníků na internetu jsou s ním velmi dobré zkušenosti (přesnost, dlouhodobá funkčnost). Princip měření výrobce neuvádí, dokonce i v často kladených otázkách odpovídá na otázku principu měření:

*„We get a lot of inquiries about how the probe works. All we can say is that it's - magic. ;)”*[19]

Ovšem po zhodnocení známých indicií předpokládám, že princip je založen na měření kapacity za použití vysokých frekvencí (desítky Mhz). Měření je přesné a časově stálé, není nutné s časem rekalibrovat, měřící hrot neobsahuje kovy podléhající korozi a také nijak neovlivňuje substrát (zasolování působením elektrolýzy). Také reakční doba je velice dobrá, výrobce uvádí, že za 1 s je již výstup z čidla stabilizován. Křivka závislosti vlhkosti na výstupním napětí by měla být proporcionální.

Výrobce také tvrdí, že je čidlo levné. Je sice levnější než srovnatelná konkurence, ale s cenou **37,95 USD** mi stále přijde relativně drahý pro mé použití a rozhodl jsem se pro konstrukci čidla vlastního. V případě, že by se v budoucnosti objevily problémy s čidlem mé konstrukce, použil bych toto čidlo.

### 3.2.4 VIRRIB

<<http://www.amet.cz/virribcz.html>>



Obrázek 9: Snímač VIRRIB

Snímač půdní vlhkosti VIRRIB pracuje na principu fázového posunu (Phase Transmission). Je určen pro zakopání do různých hloubek, například na poli apod. Jeho výstup je proveden proudovou smyčkou 0-5mA nebo napěťový 0-2,5 V, (popř. jiný dle požadavku). Rozsah měření: 5-50 % obj. vlhkosti. Přesnost měření: menší než  $0,01 \text{ m}^3 \cdot \text{m}^{-3}$ .

Snímače VIRRIB jsou vyráběny v několika variantách, a to buď jako kruhové, o průměru 28 cm, anebo podlouhlé, o šířce 6 cm a délce 20 cm. Pro potřeby měření a regulace závlahy v kontejnerovnách byl vyvinut snímač o délce elektrod 10 cm, umístitelný již do kontejneru o objemu 3 litrů. Ve všech případech jsou aktivní části provedeny z nerezové oceli a jsou fixovány v plastové skříňce, která obsahuje elektronické obvody. Snímač je nerozebíratelný.

Po recalibraci je snímači VIRRIB možno měřit objemovou vlhkost i jiných substrátů (např. písek, štěrk, obilí, uhlí apod.)

Cena: **1200 Kč + DPH**

### 3.2.5 Vodivostní senzor vlastní konstrukce

Po zhodnocení situace na trhu s čidly jsem se rozhodl pro konstrukci vlastního čidla. Skládá se ze dvou plechových hrotů z nerez, tvarem jsem se inspiroval u čínského senzoru (3.2.1). Plechy jsou



Obrázek 10: Senzor vlastní konstrukce bez izolace

provrtány a sešroubovány s cuprexitovou destičkou, ke které jsou připájeny přívodní vodiče, ty jsou taktéž připájeny k pájecím očkům přichyceným z druhé strany plechu pod hlavičky šroubů, pro jistotu skutečně dobrého spojení vodiče s nerezovým plechem. Celá tato konstrukce je zalita tavnou pistolí a povrch následně sjednocen zapalovačem.

Měří se vodivost střídavým proudem, čímž jsou eliminovány účinky elektrolýzy oproti měření stejnosměrným proudem. Elektrody jsou zasunuty přímo do substrátu, čímž získáme dobrou reakční dobu, avšak výsledky měření se budou lišit v závislosti na množství soli v substrátu. Zvolil jsem tuto variantu, protože je velmi levná a zajímalo mne, jakou budou mít životnost nerezové elektrody, jelikož teoreticky by neměly korodovat. Jedinou nevýhodou by tedy měla být nemožnost přesné a obecné kalibrace měřené hodnoty na množství závlahy.

Kalibrace se musí provést pro každé konkrétní použití, hodnoty se budou lišit podle množství soli v konkrétním substrátu. Uživatel si tedy musí sám naměřit hodnoty podle kýžených hladin vlhkosti půdy. Nepředpokládám že by laický uživatel věděl jaké hodnoty objemové vlhkosti potřebuje nastavit a stejně bude systém nastavovat tzv. „od oka“. Pokud by bylo potřeba měřit přesné hodnoty objemové vlhkosti musel by se použít senzor pracující na principu TDR, či rozdílu kapacity dielektrika. Návrh jak by se dal takový senzor vyrobit je uveden v následující kapitole (3.2.6).

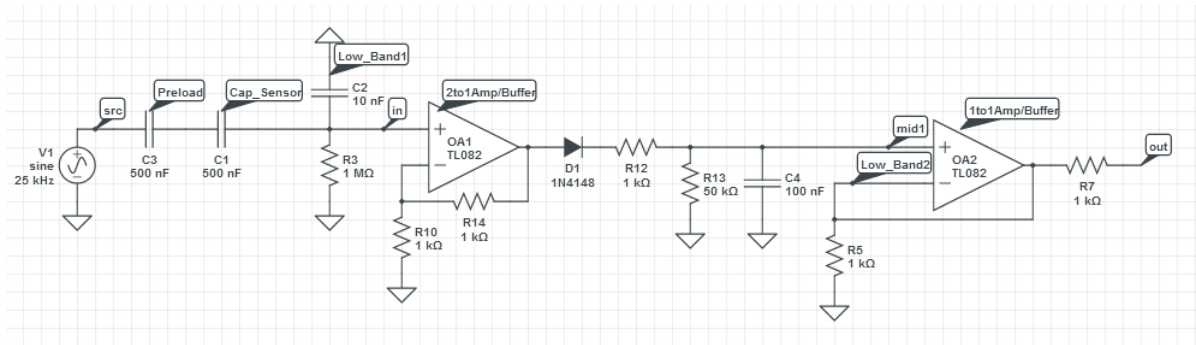
### 3.2.6 Kapacitní senzor vlastní konstrukce



Obrázek 11: Kapacitní senzor podle DrX

Velmi zajímavou konstrukci senzoru pracujícího na principu změny kapacity v závislosti vlhkosti dielektrika - substrátu uveřejnil blogger s přezdívkou DrX na svém blogu [6]. Využívá desky plošných spojů (viz. obr. 11), na které je vyleptán tvar kapacitního senzoru a celý je přelakován nevodivým lakem. Dává volně k dispozici i zdrojové soubory plošného spoje ve formátu Eagle. Spoj si nechal

vyrobit u specializované firmy a poté provedl test savosti ve sklenici s vodou, kde senzor uspěl - i po 12-ti hodinách ve vodě jeho odpor přesahoval nejvyšší rozsah použitého měřícího přístroje - 10 M $\Omega$ .



Obrázek 12: Zapojení obvodu pro kapacitní senzor[12]

V diskusi pod tímto příspěvkem je také uveřejněno několik schémat zapojení měřícího obvodu pro tento senzor, např. to z obr. 12 inspirované zapojením z <tuxgraphics.org>[14], které poskytuje stejnosměrné napětí na výstupu odpovídající měření vlhkosti. Bohužel jsem na tento blog narazil až po sestrojení zařízení a příliš pozdě na to, abych tento senzor sestrojil a vyzkoušel, ani jsem to nepovažoval za nutné. Ovšem je možné tento senzor připojit k mnou navržené řídicí jednotce k druhému vstupnímu konektoru.

### 3.3 Výběr akčního členu

#### 3.3.1 Čerpadlo



Obrázek 13: Čerpadlo ostřikovačů LADA s bazalkou

Na trhu je nepřehledné množství čerpadel vhodných k potřebám automatické závlahy. Já jsem se zaměřil na menší čerpadla s bezpečným napájecím napětím 12 V. Tyto čerpadla je možné napojit přímo na řídicí jednotku a v případě potřeby celé zařízení napájet z 12 V baterie. Malé akvariijní

čerpadla začínají na cenách lehce přes 100 Kč a pro zalévání květináče je takové čerpadlo dostatečné. V nabídce jsou také silnější čerpadla určená pro karavany, nebo i čerpadla s vysokým výkonem, avšak v jiné cenové relaci.

Pro sestavení prototypu závlahového systému jsem zvolil čerpadlo ostřikovačů vymontované z dosloužilého automobilu značky Lada. Napájecí napětí je 12 V a má vlastní nádržku. Je možné ho připojit přímo k řídicímu systému. Proud jsem při připojení na 12 V zdroj naměřil 1,5 A, při rozběhu špičkově necelé 2 A<sup>3</sup>, což je sice v porovnání s běžným akvarijním čerpadlem schopným pracovat s proudem kolem 100 mA [14] poměrně hodně, ale získal jsem ho zdarma a nenapájím systém z baterií.

### 3.3.2 Elmag. ventil



Obrázek 14: Trojitý Elmag. ventil z pračky

Pro instalace s přívodem vodovodního řádu se nabízí možnost připojení elektromagnetického ventilu, který by otevíral přívod vody do trysek pro zalévání. Výhody tohoto řešení jsou: není potřeba doplňovat nádrž s vodou, dobrý tlak pro lepší rozprášení vody, přesné dávkování. Nevýhoda může být při poruše zařízení, kdyby došlo k trvalému otevření ventilu, hrozilo by vyplavení objektu. Zajisté by v tomto případě bylo dobré použít nějaké nezávislé bezpečnostní zařízení zajišťující odpojení přívodu vody v případě poruchy.

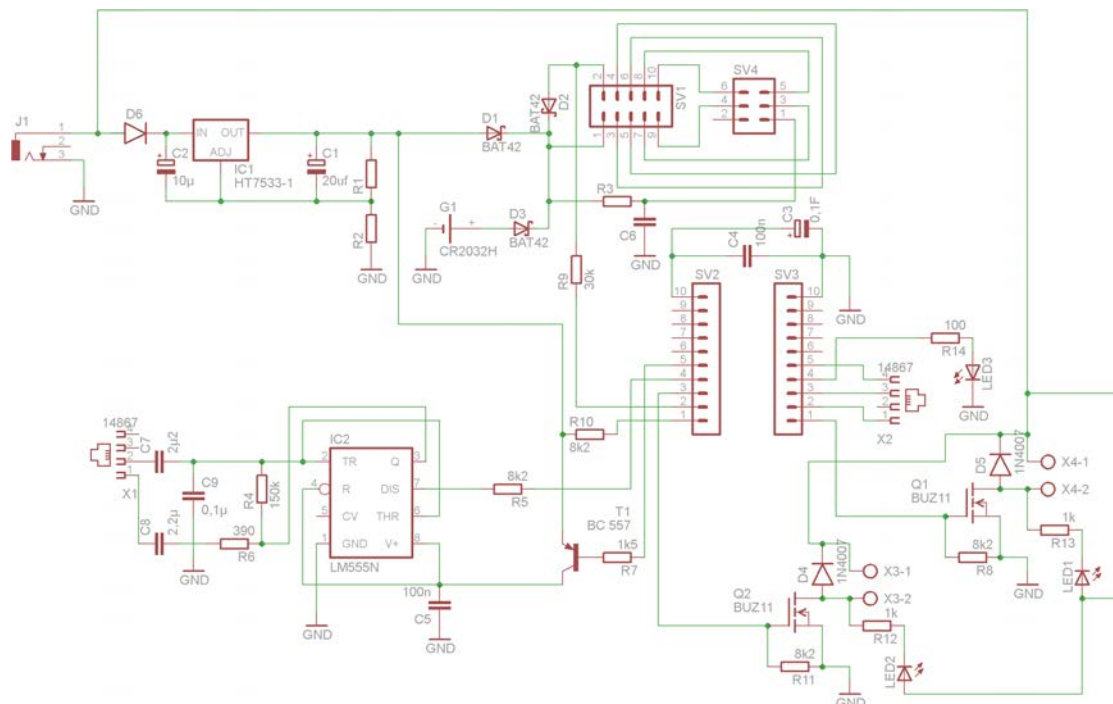
Na trhu existuje řada komerčně prodávaných ventilů k tomuto použití. Já jsem ovšem vymontoval z dosluhující pračky krásný trojitý ventil se vstupem na standardní hadici s aqua-stopem a třemi výstupy (obr. ), každý se samostatným elektromagnetickým ventilem na 230 V 50 Hz. Tudíž pro připojení k řídicí jednotce by tyto ventily musely být spínány například pomocí elektromagnetických relé.

## 3.4 Návrh řídicí jednotky

### 3.4.1 Schéma řídicí jednotky

<sup>3</sup>Hodnoty proudu jsou omezeny použitým zdrojem, při provozu z autobaterie čerpadlo spotřebovává ještě více energie a podává i vyšší výkon, ale to v tomto případě není žádáno - hrozí vyplavení zeminy.





Obrázek 15: Schéma řídicí jednotky

Z důvodů uvedených v kapitole 3.4.6 jsem se rozhodl zařízení sestavit jako tzv. boosterPack - rozšiřující desku pro launchpad, která se na něj nasadí a spolu dohromady tvoří funkční prototyp. Schéma zapojení zmíněné rozšiřující desky je na obr. 15. Propojení s launchpadem zajišťují kolíkové zásuvky SV1, SV2 a SV3. Zatímco SV2 a SV3 jsou na launchpadu zapojeny přímo na 20-pin patici mikroprocesoru, SV1 je nezvykle napojena na blok pinů oddělující mikroprocesor od programátoru/debuggeru, USB převodníku a stabilizátoru napětí zajišťujícím napájení mikroprocesoru při zapojení launchpadu do USB.

Schéma je kompatibilní s MSP430 LaunchPad ve verzi 1.4. V průběhu vývoje se mi omylem podařilo tento vývojový kit rozbít. Jelikož dnes se prodává Launchpad ve verziá jinak zapojený konektor SV1, byl jsem nucen upravit plošný spoj pro použití s LaunchPadem verze 1.5. Navíc odpor R3 a kondenzátor C6 zajišťující potřebné zpoždění napětí na RST pinu po zapnutí, není potřeba osazovat, protože LaunchPad verze 1.5 již tyto součástky obsahuje.

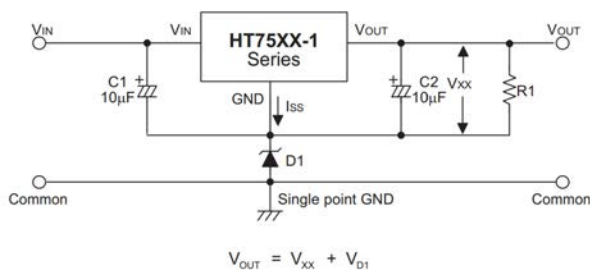
Schémata zapojení obou verzí LaunchPadu jsou v přílohách.

### 3.4.2 Napájení zařízení

Zařízení může být napájeno buď z externího zdroje 6 - 24 V přes konektor J1, nebo z USB portu na launchpadu. Jelikož na zařízení běží hodiny reálného času zajištěné samotným mikroprocesorem, je na desce umístěna i záložní 3 V baterie G1 dodávající napětí pro běh hodin při ztrátě obou dalších napájení. Všechny tři zdroje napájení jsou propojeny shottkyho diodami D1 - D3, aby bylo zamezeno jejich vzájemnému ovlivňování, především pak nekontrolovanému nabíjení (zvolený typ záložní baterie není nabíjecí), či nechtěnému vybíjení záložní baterie.

Zkoušel jsem i místo baterie použít záložní kondenzátor C3 o hodnotě 0,1 F, avšak ten vydrží zálohovat hodiny po dobu pouhých dvou dní (což potvrdil i kontrolní výpočet, který jsem provedl později), proto jsem místo něj zvolil záložní baterii, která by měla vydržet zálohovat napájení v řádech let. Odběr ze záložní baterie se mi nakonec podařilo snížit na hodnotu 2,5  $\mu\text{A}$ , což považuji za úspěch, ačkoliv výrobce mikroprocesoru hovoří o spotřebě 1  $\mu\text{A}$  v Low Power Mode 3, který používám (tzn. během čekání mezi sekundami). Při takto nízkém odběru se ale stačí dotknout prstem desky plošného spoje v místě, kde není žádný vodivý kontakt, či přiblížit se k vodičům od měřicího přístroje rukou a měřená hodnota je rázem zkreslená o jednotky  $\mu\text{A}$ . V takto nízkém měřicím rozsahu nebude ani přesnost levného měřicího přístroje vysoká.

Detekce ztráty napájení je zajištěna hlídáním napětí před oddělovacími diodami. Na vstupní piny mikroprocesoru je přivedeno pomocí odporů R9 pro USB a R10 pro externí zdroj, společně s interními pull-down rezistory tvoří odporový dělič snižující mírně vstupní napětí na vstupech, pro jejich ochranu před nadměrným napětím. S tímto zapojením se ale vyskytl problém. Pokud je zařízení připojené zároveň k USB i externímu zdroji a následně odpojíme USB, celá část launchpadu s USB převodníkem, debuggerem atd. zůstane napájena přes RXD a TXD piny sériového kanálu z externího zdroje. Napětí v této části ale poklesne z 3,6 V na 2 V, využil jsem tohoto vedlejšího efektu a spočítal odporový dělič tak, aby při 2 V bylo napětí na děliči nižší než spodní hraniční napětí pro log. 1 (1,47 V) a při 3,6 V vyšší než horní hranice pro log. 0 (1,8 V)<sup>4</sup>. Vychází to o 0,273 V víc/méně než jsou hranice, což není mnoho, ale funguje to spolehlivě. Při konstrukci finálního výrobku bych ale doporučoval vycházet z napětí 5 V poskytovaným přímo USB připojením, kde tento problém nehrozí. Při detekci ztráty USB napájení ihned softwarově změním funkci RXD a TXD pinů na digitální vstupy, čímž celou „USB část“ LaunchPadu „odříznu“ od napájení, tato pak nespotřebává zbytečně energii a nehrozí její poškození.



Obrázek 16: Navýšení výstupního napětí stabilizátoru diodou[8]

Napájecí napětí mikroprocesoru při použití externího zdroje musí být dostatečně vyšší než napájecí napětí záložní baterie (aby nedocházelo k jejímu vybíjení během provozu). S ohledem na úbytek napětí na oddělovací diodě D1 (přibližně 0,3V u zvoleného typu při provozním odběru proudu) jsem zvolil výstupní napětí stabilizátoru 3,6 V. Takový stabilizátor se bohužel v obchodech se součástkami v České republice neprodává, proto jsem pro potřeby prototypu použil stabilizátor napětí HT7533-1, s výstupním napětím 3,3 V, toto napětí jsem měl v plánu navýšit zařazením odporového děliče tvořeným odpory R1 a R2 jak je vidět na schématu (obr. 15). Vedly mě k tomu obavy z teplotní stability katalogového zapojení pro navýšení výstupního napětí diodou (viz. obr. 16). Avšak zapojení

<sup>4</sup>Hraniční napětí logických úrovní jsou určeny procentně z napájecího napětí a rozsahy pro log. 0 a 1 se částečně překrývají

s odporovým děličem by spotřebovávalo řádově více energie a to by při případném provozu z baterií bylo škoda. Proto jsem nakonec zvolil zapojení (obr. 16) s diodou 1N4148, které dává na výstupu 3,52 V k mému překvapení poměrně stabilně. Po proměření výstupního napětí tohoto zapojení v při různých teplotách jsem usoudil, že přesnost na 0,1 V je pro potřeby měření více než dostatečná. Zařízení tak v idle režimu spotřebovává skutečně zanedbatelný proud (kolem 0,1 mA). Pro ochranu zařízení při přepólování externího zdroje jsem před stabilizátor zařadil diodu D6 (15).

### 3.4.3 Vstupy pro senzory

Chtěl jsem porovnat výsledky měření Čínského čidla (kapitola 3.2.1) s čidlem mé konstrukce (kapitola 3.2.5). Proto jsem zařízení opatřil dvěma vstupy, každý pro jeden typ senzoru. Konektor X2 určený pro Čínský senzor má zapojeny 3 kontakty přímo do mikroprocesoru, čímž je zajištěna flexibilita tohoto vstupu. V případě potřeby je do tohoto vstupu možné napojit i jiné čidlo, které má napěťový výstup s rozsahem do 3 V. Po úpravě firmware zařízení je možné připojit senzor s frekvenčním výstupem. Důvod proč jsou i napájecí piny GND a VCC pro Čínský senzor napojeny na I/O piny mikroprocesoru je ten, že jsem chtěl použít dvě vodivostní čidla v jednom květináči, abych mohl porovnat výsledky jejich měření. Aby se tyto dva senzory navzájem neovlivňovaly, po dobu měření senzorem mé konstrukce jsou napájecí piny Čínského senzoru nastaveny jako digitální vstupy, ty tak nejsou spojeny se zemí, ani s napájecím napětím a měření druhým čidlem tak není ovlivněno. Možnost odpojení senzoru od napájení po dobu kdy na něm není prováděno měření je u tohoto senzoru více než nutnost (stejnoseměrný proud procházející měřicími hroty po dobu měření způsobuje jejich oxidaci).

Konektor X1 má zapojeny pouze dva piny určené pro dva nerezové hroty senzoru mé konstrukce. Zařízení pak obsahuje měřící obvod pro měření střídavým proudem popsany v následující kapitole (3.4.4). Hroty jsou galvanicky oddělené kondenzátory, tudíž senzor neovlivňuje výsledky měření Čínským senzorem. Pro oba vstupy jsem zvolil konektor RJ-11 známý z telekomunikační techniky.

### 3.4.4 Měřící obvod

Pro měření vodivosti střídavým proudem jsem zvolil osvědčené zapojení<sup>[7]</sup> se slavným obvodem 555 v CMOS verzi (funguje při 3 V napájení, stejně jako použitý mikroprocesor MSP430). Zapojení funguje jako oscilátor, který nabíjí a vybíjí fóliový kondenzátor C9 o hodnotě 0,1  $\mu\text{F}$  částečně přes měřicí elektrody galvanicky oddělené bipolárními kondenzátory C7 a C8 o hodnotách 2,2  $\mu\text{F}$ , čímž je zajištěn skutečně střídavý průběh, nikoliv jen tepavý<sup>5</sup>. Sériově s měřicími elektrodami je zapojen odpor R6 o hodnotě 390  $\Omega$ , který určuje maximální výstupní frekvenci při zkratovaných elektrodách<sup>6</sup>. Paralelně k elektrodám je napojen odpor R4 o hodnotě 150 k $\Omega$ , který určuje minimální výstupní frekvenci při rozpojených elektrodách - odpojené čidlo, či elektrody nejsou v substrátu.

<sup>5</sup>Můj první pokus o měření střídavým proudem vedl přes využití interního komparátoru mikroprocesoru, avšak bez galvanického oddělení kondenzátory a výsledky měření nebyly stabilní, důsledkem působení elektrolýzy jedním směrem. Při použití skutečně střídavého proudu je tento jev eliminován.

<sup>6</sup>Ve skutečnosti může být frekvence i o něco vyšší než při zkratovaných elektrodách, například při ponoření elektrod do solného roztoku.

### 3.4.5 Výstupy pro akční členy

Výstupy pro čerpadla, či vodovodní ventily jsem vyřešil pomocí konektorů s odnímatelnou svorkovnicí, kam je možné napojit vodiče přímo od čerpadla, které je na stejné napětí jako externí napájecí zdroj. Další možností je zapojení elektromagnetického relé pro spínání akčních členů na síťové napětí 230 v 50 Hz, například elektromagnetický ventil zmíněný v kapitole 3.3.2. MOSFET tranzistory BUZ 11 spínají zem v tomto konektoru a druhý pin je napojen přímo na napájecí napětí z externího zdroje. Oba výstupy jsou doplněny signalizační LED diodou a také antiparalelně zapojenou diodou, která chrání spínací tranzistory proti nežádoucím napěťovým špičkám vznikajícím při indukční zátěži, což čerpadlo a elektromagnetické relé jsou.

### 3.4.6 Komunikace s PC

Zvolil jsem s ohledem na požadavek komunikace s PC mikrokontrolér MSP430G2553, který obsahuje hardwarový UART. Dnešní počítače již většinou nejsou vybaveny sériovou linkou, tudíž je potřeba užít převodníku na USB, které je dnes standardem mezi osobními počítači. Máme na výběr z mnoha možností.



Obrázek 17: FTDI modul

Existují převodníky ve formě redukce z USB na RS-232, ovšem zařízení vybavené přímo USB portem je mnohem elegantnější. Za tímto účelem je nutné zařízení vybavit samotným převodníkem. Dá se použít například čip FTDI, který toto zařizuje a k němu patří různé součástky, nebo se na internetu prodávají hotové převodníky s tímto čipem na destičce s USB konektorem (viz. 17), které se jen připojí k vašemu zařízení. Pro stavbu prototypu se toto řešení zdá výhodné, avšak já se rozhodl použít pro potřeby prototypu celý *LaunchPad* tak jak je a zařízení pojmout formou tzv. „booster packu“ tzn. destičky která se nasadí na *LaunchPad* a tím vznikne prototyp celého zařízení. Hlavní výhodou, proč jsem zvolil toto řešení je mimo vyřešení USB komunikace také jednoduchá možnost aktualizace firmwaru a debugingu pomocí stejného USB kabelu. Ve fázi vývoje zařízení se předpokládá časté ladění a obměna firmwaru, která je takto šikovně vyřešena bez nutnosti externího programátoru.

V současné době již začínají být rozšířené mikroprocesory schopné komunikovat přes USB přímo bez nutnosti převodníku. V případě sériové výroby by bylo použití takového mikroprocesoru levnější než zapojení s převodníkem. Avšak k programování těchto mikroprocesorů by bylo zapotřebí mnohem dražšího vybavení, než je *LaunchPad*, určený pro řadu ValueLine<sup>7</sup>, tudíž jsem o tom ani neuvažoval.

### 3.4.7 Deska plošného spoje

Při tvorbě prototypu jsem dal přednost použití klasických elektronických součástek před technologií SMD, mohl jsem tak použít součástky již použité v kontaktním poli, na kterém jsem vývoj hardwaru počal. V sériové výrobě samozřejmě vychází použití technologie SMD, součástky v tomto provedení bývají levnější a navíc není potřeba vrtat díry.

<sup>7</sup>Podle neoficiálních zdrojů[15] lze *LaunchPad* použít i k programování procesorů mimo řadu ValueLine.

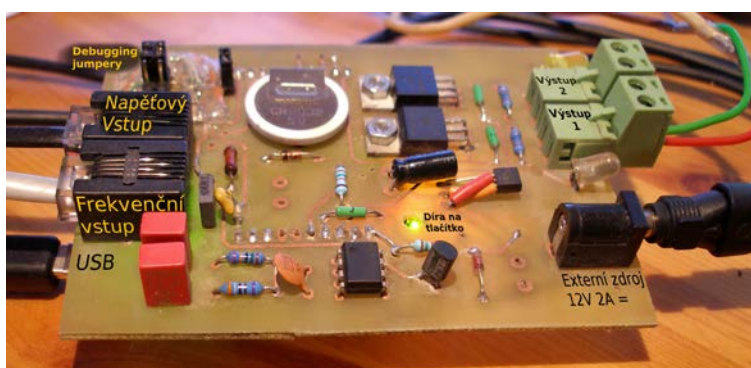
Díky použití klasických součástek, především pak konektorů rj-11 (které jsou moc vysoké na to aby mohly být ze spodní strany) a koncepcí zařízení jako *Booster Pack* k *LaunchPadu*, kde je potřeba ze spodní strany plošného spoje osadit kolíkové zásuvky jsem byl nucen vyrobit plošný spoj oboustranný.

Plošný spoj jsem navrhoval v programu EAGLE, který je do určité velikosti spoje zdarma. Na radu kamaráda jsem použil metodu nažehlování toneru, kdy jsem vytiskl předlohu z Eaglu na letákový papír, přezehlil ji na cuprexit, odmočil a odstranil papír, že zůstal jen toner na mědi a spoj vyleptal v chloridu železitém. Druhou stranu jsem provedl obdobným způsobem s tím, že jsem vyvrtal centrovací dírky do spoje a usadil na ně přes vrtáčky papír pro druhou stranu, po přesném usazení je potřeba z druhé strany přilepit předlohu k destičce aby se nemohla pohnout. Při leptání jedné strany jsem druhou přelepil lepicí páskou pro její ochranu před leptací kapalinou. Více informací s fotografiemi k této metodě se dá najít například na tomto blogu: [http://thomaspfeifer.net/direct\\_toner\\_pcb.htm](http://thomaspfeifer.net/direct_toner_pcb.htm).



Obrázek 18: Detail DPS

### 3.4.8 Hotový prototyp hardware



Obrázek 19: Pohled na vrchní desku zařízení

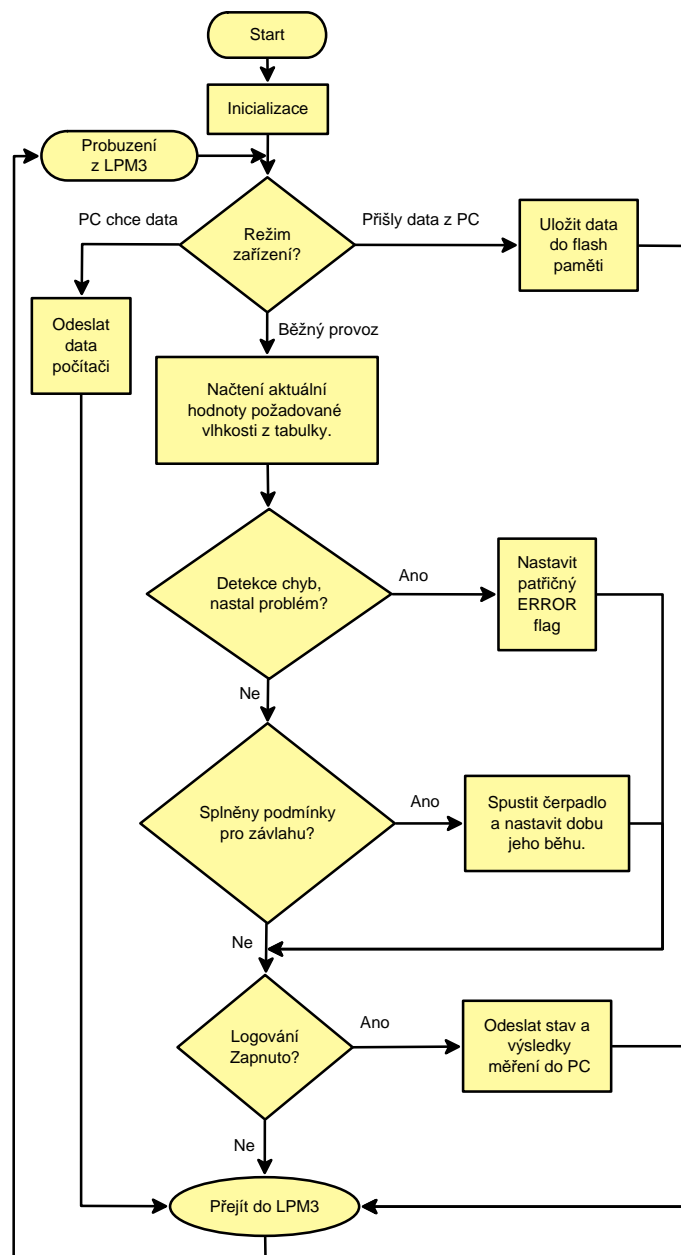
Na obrázku 19 je vidět osazená deska plošného spoje z vrchní strany s popisem klíčových prvků. Tato deska je nasunuta na *LaunchPad* který ji prosvětluje zespodu svými indikačními LED diodami. Zároveň je v desce díra v místě, kde je na *LaunchPadu* umístěno tlačítko, aby bylo možné toto tlačítko něčím stisknout skrz tuto díru. USB port je rovněž umístěn na spodní desce - *LaunchPadu*. *Debugging jumper* slouží pro propojení mikroprocesoru s debuggerem na *LaunchPadu* - při jejich spojení jako na obrázku je možné upravovat firmware zařízení a také jej debuggovat přímo v zapojení, druhá poloha slouží jen k jejich odložení v případě, že chceme mít programátor/debugger odpojen, to je potřeba pokud je zařízení připojeno jen na externí zdroj.<sup>8</sup>

<sup>8</sup>Jedná se o vodiče sběrnice Spy-Bi-Wire, která využívá piny mikroprocesoru TEST a RST, pokud bychom je nechali připojené k *LaunchPadu* při provozu pouze na externí zdroj, blokový RST pin by nedovolil mikroprocesoru se spustit. Proto je v tomto případě zapotřebí tuto sběrnici odpojit pomocí jumperů .

## 4 Firmware mikroprocesoru

Programové vybavení mikroprocesoru jsem vytvářel ve vývojovém prostředí *Code Composer Studio* verze 5.5 (vychází z prostředí *Eclipse*). Firma TEXAS INSTRUMENTS jej nabízí volně ke stažení. S omezením velikosti kódu i licenci zdarma. Program jsem psal v jazyce C, pro který jsou mikroprocesory MSP430 optimalizovány [16].

### 4.1 Hlavní smyčka programu



Obrázek 20: Vývojový diagram hlavní programové smyčky

Program je navrhován v souladu s pravidly pro nízký odběr energie. Většinu času tráví v úsporném

režimu *LPM3*, ze kterého jej probouzí obsluha přerušování, pokud je potřeba provést něco z nekonečné smyčky. Po provedení jedné iterace nekonečné smyčky se mikroprocesor opět uvede do úsporného režimu a čeká na další probuzení.

Probuzení z úsporného režimu nastane buď na popud přerušování hlavního časovače (pro logování dat do PC, nebo nastal čas zkontrolovat, zda-li není potřeba zavlažit zeminu, či provést kontrolní měření), nebo na popud přerušování od sériového kanálu *UART* (jsou připraveny data pro zapsání do paměti *flash*).

Při běžném provozu se nejdříve v tabulce hodnot (data křivky vlhkosti) najde hodnota platná pro aktuální čas (pokud nejsou data v tabulce v pořádku, nebo chybí, vyhlásí chybu a nebude zalévat). Poté se ověří stav senzoru a případně vyhlásí jinou chybu (zkratovaný senzor, odpojený senzor/senzor mimo zeminu), pokud nenastane žádná chyba, ověří se, jestli nastal čas na závlahu a vlhkost zeminy klesla pod úroveň kdy je nutné zavlažit. Podle rozdílu žádané a skutečné hodnoty půdní vlhkosti se nastaví doba běhu čerpadla/ventilu (tyto konfigurační hodnoty se budou lišit podle prostředí a je potřeba je odladit pro konkrétní situaci, aby regulace fungovala správně) a spustí se čerpadlo. Čerpadlo poté vypne hlavní časovač po uplynutí doby závlahy.

Několik minut po zavlažování se provede kontrolní měření, které má za úkol zjistit, jestli se hodnota půdní vlhkosti zvýšila. Pokud by se dostatečně nezvýšila, vyhlásí zařízení chybu a začne nepřetržitě blikat. Možné příčiny této chyby jsou:

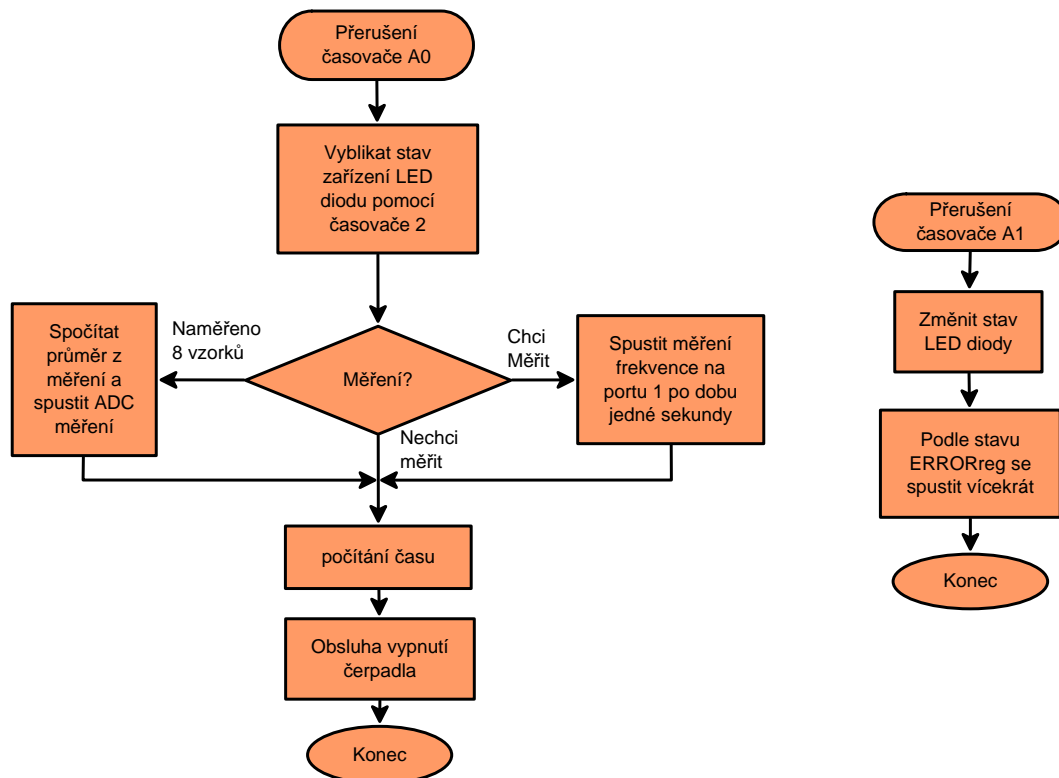
- došla voda v zásobníku
- přívodní hadice je poškozena
- hadice vypadla z květníku, nebo je rozpojena
- porucha na čerpadle, nebo spínacím obvodu
- porucha na čidle, čidlo je v jiném květníku, atd.

Jak je vidět možností je více a některé jsou kritické, například hrozí vyplavení bytu, poškození čerpadla, či dokonce požár zařízení. Proto tato chyba nebude sama vymazána uživatel ji musí po zkontrolování zařízení smazat stisknutím tlačítka na *LaunchPadu*, poté bude zas zařízení pokračovat ve své funkci (tedy pokud se chyba neobjeví znovu při dalším pokusu o závlahu).

Nakonec, pokud je *LaunchPad* připojen k počítači a je zapnuto logování, se odešlou výsledky měření spolu s informacemi o čase nastaveném v zařízení, aktuálně požadované hodnoty, stavu chybového registru, či zda bylo zavlažováno a jakou měrou. Tuto zprávu odesílá zařízení každou minutu.

## 4.2 Přerušování časovače

Časovač *A0* je nejdůležitější pro běh systému. Jeho obsluha přerušování se spustí každou sekundu, aby se aktualizoval čas v zařízení, tato část pracuje i tehdy, je-li zařízení odpojeno od externího napájení i od zdroje napájení ve formě USB.



Obrázek 21: Vývojový diagram přerušování časovače 1 a 2

Každou druhou sekundu časovač *A0* rozsvítí indikační diodu a spustí časovač *A1*, který je nastavený na dobu bliknutí diody. Pokud v zařízení nenastal žádný problém (errorReg neobsahuje příznak o chybě), časovač *A1* zhasne indikační diodu a vypne sám sebe - dioda blikne jen jednou. Pokud nastal problém, podle druhu chyby se časovač *A1* spustí vícekrát - dioda blikne vícekrát, nebo bliká nepřetržitě.

Pokud je zařízení napojeno na externí zdroj a nastal čas pro měření, časovač *A0* spustí po dobu jedné sekundy měření počtu pulzů na měřícím pinu, takto postupně naměří 8 vzorků a z nich vypočítá průměr rotováním o tři bity.<sup>9</sup>

V momentě, kdy je měření frekvence dokončeno, spustí se měření pomocí A/D převodníku na druhém vstupu - napěťovém.

Nakonec časovač *A0* každou sekundu kontroluje zda není zapnuté čerpadlo a jestli jej nemá vypnout. Při závlaze se v hlavní programové smyčce zapne čerpadlo a nastaví doba jeho běhu. Zde se doba běhu odečítá (každou sekundu) a pokud klesne na hodnotu 1, dojde k vypnutí čerpadla. Pokud je v idle stavu doba běhu čerpadla 1 a errorReg je roven nule, znamená to, že systém již zaléval a chyba nenastala - je vše v pořádku.

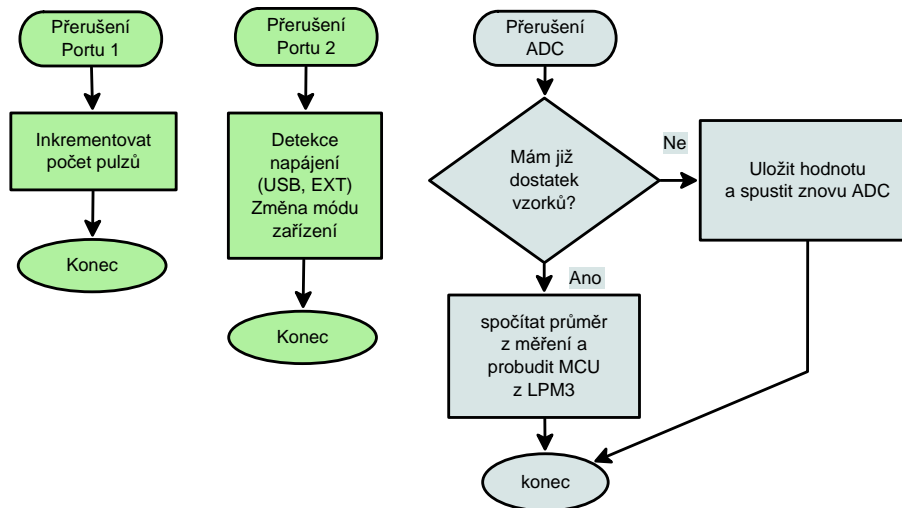
Záměrně jsem nepoužil časovač watchdog. To proto abych mohl v testovacím provozu zpozorovat případné zatuhnutí systému. Systém sice po celou dobu testovacího provozu fungoval bezvadně a ani jednou nezatuhl, ale pro bezpečnost celého systému by bylo rozhodně dobré použít watchdog [5], který

<sup>9</sup>Později jsem zjistil, že u měření frekvence touto metodou je počítání průměru z osmi vzorků zbytečné, systém podává stejně dobré výsledky i při měření jednoho vzorku. Nechal jsem jej ale měřit 8 vzorků každou minutu v logovacím módu po dobu více než měsíce, abych otestoval odolnost nerezových elektrod. Měření v běžném provozu probíhá v nastaveném intervalu, já nastavil 30 minut, po této době je již voda v zemině dobře vsáknuta.



by v případě zatuhnutí restartoval systém. Takové zatuhnutí systému během právě probíhající zvlahy by mohlo mít fatální následky.

### 4.3 Přerušeni GPIO



Obrázek 22: Vývojový diagram přerušeni od vstupních pinů a A/D převodníku

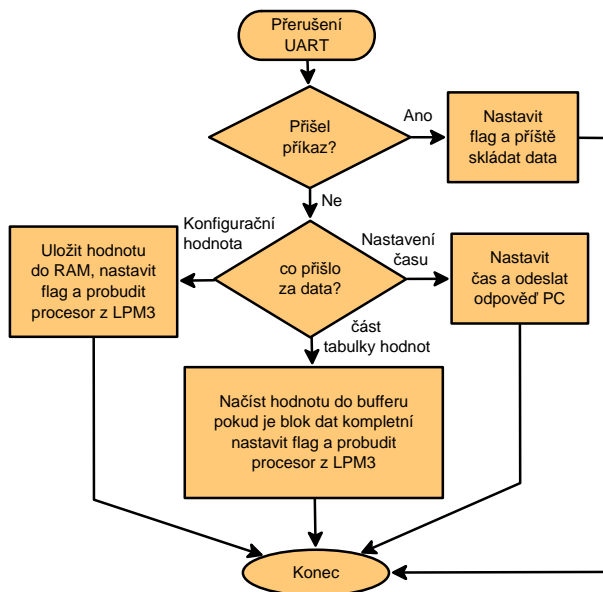
Na *portu 1* se nachází pin, pro měření frekvence produkované obvodem 555. Časovač vynuluje proměnnou a spustí měření frekvence, obsluha přerušeni portu 1 inkrementuje proměnnou pokaždé se změni stav měřícího pinu. Po uplynutí jedné sekundy časovač ukončí měření. Hodnota v proměnné je poté rovna počtu změn pinu po dobu jedné sekundy a to je měřená frekvence.

Po změření frekvence započne měření napětí pomocí A/D převodníku (ADC). Pokaždé, když ADC změří napětí, nastane přerušeni ADC. V obsluze tohoto přerušeni se opět postupně sčítá 8 měřených hodnot, ze kterých se vypočítá průměr rotováním o 3 bity (pro mikroprocesor je to rychlejší, než dělení osmi). Po vypočítání průměru se probudí mikroprocesor z úsporného režimu a data se zpracují v hlavní programové smyčce.

Na *portu 2* se nachází detekční piny napájecích napětí. *Bit 1* hlídá napájecí napětí USB portu a *Bit 2* hlídá napájecí napětí externího zdroje. Pokud se změni stav na jednom z pinů spustí se procedura

### 4.4 Přerušeni UART

Hardwarový UART vyvolá přerušeni, když přijme celý byte. Vzhledem k tomu, že potřebuji přenášet čísla, která se vejdu do 14-ti bitů, rozdělil jsem je při komunikaci na dvě části a to horních 7 bitů a spodních 7 bitů. Čísla větší než 128 (osmý bit v bajtu roven 1) slouží jako příkazy - udávají, co se bude přijímat. Výhoda je, že pokud zařízení z jakéhokoliv důvodu přijme poškozená, či nekompletní data, nečeká zařízení na dokončení přenosu. Jakmile se přijme příkaz, začne ihned od znova, nezáleží na tom v jakém stavu bylo zařízení doposud. Dalo by se to řešit i timeoutem, ale toto mi přišlo jednodušší, rychlejší a jistější.



Obrázek 23: Vývojový diagram přerušení UART

Pokud přijde z počítače příkaz, nastavím příznak říkající co se bude přijímat za data. Podle příznaku se poté nakládá s dalšími přijatými bajty. Nastavení času probíhá ihned. Operace vyžadující zápis do paměti flash se provádí v hlavní programové smyčce. Obsluha přerušení UART po nasbírání dostateku bajtů (naplnění bufferu) nastaví příznak určující, co se má udělat a probudí mikroprocesor z úsporného režimu. Proč se samotný zápis do flash paměti děje v hlavní smyčce vysvětlují v kapitole 4.5.

## 4.5 Zápis do flash paměti

Mikrokontroléry MSP430 jsou založeny na Von Neumanově architektuře. To znamená, že používají stejnou paměť pro program a pro data. Mnou zvolený mikrokontrolér MSP430G2553 používá paměť typu flash o velikosti 16 kB. Do této paměti dokáže mikrokontrolér sám zapisovat. Z principu tohoto typu paměti vyplývá, že je možné zapisovat pouze „jedničky“. Paměť je ale organizovaná do bloků (v hlavním prostoru jsou to bloky o velikosti 512 B) a tyto bloky je možné jednotlivě vymazat. Tudíž pokud chceme zapsat data, je nejdříve nutné vymazat celý blok - 512 B a poté do něj zapsat data (jedničky).

Jelikož ve stejné paměti je uložen i program mikrokontroléru, je nejdříve potřeba zjistit, kde končí samotný program v paměti a zapisovat až do bloku za ním. U tohoto typu mikrokontroléru začíná program na adrese 0xC000 a prostor končí na adrese 0xFFFF. Kde program končí se dá buď jednoduše vypočítat z velikosti programu, nebo se dá v *Code Composer Studio* obsah paměti stáhnout (musí být zastaven běh programu v zařízení) a poté je přesně vidět, na které adrese jsou ještě data a kde jsou už jen nuly. Adresa prvního volného bloku pro tabulku hodnot je v proměnné *addr*.

Také se musíme vyvarovat zapisování do oblasti 0xFE00-0xFFFF, kde jsou uloženy vektory přerušení. Pokud bychom tento poslední blok paměti přepsali, mikroprocesor by nevěděl kde pokračovat, kdyby nastalo přerušení[20]. V momentě, kdy zapisujeme do flash paměti také musí být zakázány veškerá přerušení. Během zápisu není paměť přístupná pro čtení a kdyby nastalo během zápisu přerušení, čelili

bychom stejnému problému jako v předchozím případě. Proto je nutné před zapisováním/mazáním přerušení zakázat, po provedení operace je můžeme opět povolit. Následuje ukázka, jak se maže blok paměti začínající na adrese *addr*.

Výpis 1: Vymazání flash segmentu

---

```

1 void flash_erase(int *addr) {
2     __disable_interrupt();           // Zakážeme přerušení
3     while(BUSY & FCTL3);           // Počkáme, až bude
4     FCTL2 = FWKEY + FSSEL_1 + FN3; // Clk = SMCLK/4
5     FCTL1 = FWKEY + ERASE;         // Mód mazání
6     FCTL3 = FWKEY;                 // Odemkneme paměť
7     *addr = 0;                     // Vymažeme segment
8     while(BUSY & FCTL3);           // Čekáme až bude operace hotova
9     FCTL1 = FWKEY;                 // Zrušíme mód mazání
10    FCTL3 = FWKEY + LOCK;           // Zamkneme paměť
11    __enable_interrupt();           // Povolíme přerušení
12 }
```

---

Po tomto smazání je možné do bloku paměti zapisovat, zapisování probíhá obdobně jako vymazání, jen místo ERASE flagu je nastaven WRT flag (mód zápisu) a místo zapisování nuly se zapíše cyklem obsah předem naplněného bufferu na po sobě jdoucí adresy *addr*.

První záznam v paměti na adrese *addr* je počet následujících záznamů v tabulce. Po něm následují 16-ti bitová čísla, nejprve čas jako timestamp v minutách a po něm hodnota vztahující se k onomu času. Jelikož během zápisu dat do paměti musí být zakázány přerušení a nebylo by dobré přijít o přerušení časovače, který počítá čas, stará se o vypnutí čerpadla atd., před samotným ukládáním dat do paměti zajistím čekací smyčkou, že na uložení dat bude dostatek času. TAR je registr časovače A0. Vše je nastaveno tak, že když tento registr dosáhne hodnoty 511, nastane přerušení časovače přesně po jedné minutě. Čekací smyčka tedy jednoduše kontroluje, jestli již hodnota registru není příliš vysoká, v takovém případě nedovolí ukládat data do paměti dřív, než proběhne přerušení časovače a hodnota TAR se vynuluje.

Výpis 2: Ukládání tabulky do paměti

---

```

1 while(TAR>300); // Počkat na správný moment
2 if(tableBlockIndex==0){ // Byl přijat první paket
3     flash_erase(addr); // Vymazat blok v paměti
4 }
5 if(tableRowCount < 64){ // Poslední paket přijat
6     // Na první místo v tabulce zapsat počet záznamů tabulky
7     flash_write_value(addr, (tableBlockIndex*64 + tableRowCount));
8 }
9 // Uložit data z paketu do paměti
10 flash_write_block(addr + 1 + tableBlockIndex*64);
```

---

Z počítače přichází postupně data v paketech po 64 hodnotách. V kódu tyto pakety nazývám *tableBlocky*, ale aby se to nepletlo s flash bloky hovořím zde o paketech. Každý paket má svůj *tableBlockIndex* první má index 0, před jeho zapsáním je nutné vymazat blok paměti flash. Poslední paket

pravděpodobně nebude mít všech 64 hodnot a tudíž ho tak poznáme a můžeme spočítat počet všech hodnot v tabulce, který následně uložíme na první adresu flash tabulky. Podle toho pak program pozná, který záznam v tabulce je poslední a kdy bude prohledávat opět od začátku. Pokud by přeci jen vyšel počet záznamů v posledním paketu na plných 64, musí po něm počítač poslat ještě jeden prázdný paket s hodnotou *tableRowCount* rovno nule, tím se zápis uzavře. V opačném případě by nebyly data v paměti kompletní a systém by hlásil chybu `ERROR_DATA` a nemohl by zalévat. Po úspěšném uložení dat do paměti o tom dá mikroprocesor vědět počítači a ten může posílat další paket.

## 4.6 Procházení tabulky v paměti

Když je v paměti uložena tabulka hodnot ve správném formátu je z ní potřeba před kontrolou vlhkosti načíst aktuálně žádanou hodnotu s kterou se bude měřená vlhkost porovnávat. To zajišťuje následující úryvek kódu.

Výpis 3: Hledání aktuální hodnoty

---

```

1 i=0; // Počet posunů v tabulce
2 while(~errorReg&ERROR_DATA){ // Pokud nastane chyba ukončit cyklus
3   timestampNow = hour*60 + min + day*1440; // Počet minut od pondělí 0:00
4   if(nextTime<currentTime) // Další hodnota nastane až příští týden
5     nextTime += 10080; // Přičteme dobu týdne v minutách (7*24*60)
6   if(timestampNow<currentTime) // Zkoumaná hodnota je z minulého týdne
7     timestampNow += 10080; // Přičteme týden k času "teď"
8   if(timestampNow>=nextTime){ // Čas další hodnoty již nastal
9     currentTime = *(addr+flashTablePosition); // current = next
10    flashTablePosition++; // Posun na hodnotu vlhkosti
11    currentValue = *(addr+flashTablePosition); // Načíst novou hodnotu
12    flashTablePosition++; // posun na další čas
13    flashTablePosition %= *addr; // Rotace z konce na začátek
14    nextTime = *(addr+flashTablePosition); // Čas další hodnoty
15    if(i>*addr) // Testuje se znovu dokola celá tabulka
16      errorReg |= ERROR_DATA; // To je chyba, vedlo by to k zacyklení
17      i++;
18    }else{ // Našli jsme hodnotu platnou nyní
19      threshold=currentValue; // Nastavíme žádanou hodnotu vlhkosti
20      break; // A ukončíme cyklus
21    }
22 }

```

---

Jde pouze o skokové načítání hodnot v čase. O dopočítávání křivky mezi hodnotami jsem se na malém mikroprocesoru ani nepokoušel, navíc to ani není potřeba, ba naopak. Uživatel si takto může zvolit přesně moment, kdy se má začít zavlažovat (například ráno, kdy to mají rostliny nejraději). Pokud by přeci jen byl žádán pozvolně rostoucí průběh, nic nebrání uložení více po sobě jdoucích hodnot za sebou. Ovšem vzhledem k povaze akčního členu (minimální dávka čerpadla), bude průběh vždy trochu „zubatý“.

## 4.7 Regulace půdní vlhkosti

Po provedení kontrolního měření následuje porovnání naměřené hodnoty s aktuálně žádanou hodnotou z tabulky. Pokud je naměřená hodnota nižší o hodnotu 500 Hz, což je prahová hodnota pro malou závlivku, spustí se závlaha po dobu nastavenou pro malou závlivku (pro malý testovací květináč jsem nastavil 2 s s čerpadlem na ostřikovače), pokud je měřená hodnota nižší než 2000 Hz, nastane velká závlivka (já zvolil 4 s). Tyto prahové hodnoty se samozřejmě dají změnit a pokud by to bylo nutné dalo by se jednoduše přidat víc časových konstant s různými prahovými hodnotami. Já to v rámci zachování jednoduchosti nepovažoval za nutné. Velmi dobrých výsledků jsem dosáhl i s tímto jednoduchým systémem.

V případě, že je zemina velmi vyschlá a je požadována vysoká půdní vlhkost nastane velká závlivka, která urychlí nárůst vlhkosti v půdě. Nemusí být ale dostatečná k dosažení požadované vlhkosti, v tomto případě bude system pokračovat v zalévání v dalším kontrolním intervalu (já nastavil 30 minut), do té doby, než bude požadavek na vlhkost splněn. Zdá se to jako nedokonalost regulace, ale naopak je to vhodné, protože zemina není schopna pojmout velké množství vody najednou a potřebuje čas na vsáknutí. Tudíž i v tomto ohledu systém regulace vyhovuje a nehrozí tak odplavení zeminy velkým množstvím vody.

Nasazení jinak hojně používané a oblíbené regulace s prvky PID, by při takto dlouhodobých jevech nedávalo příliš smysl. Byl by velký problém ji vyladit, byť i pro jedno konkrétní nasazení při použití konkrétních akčních členů, s konkrétními senzory na konkrétním místě. Navíc z nutnosti tohoto složitěho ladění pro každé nasazení vyplývá naprostá nevhodnost této kategorie regulace.

## 4.8 Detekce napájecího napětí

V kapitole 3.4.2 jsem se zmínil o úspoře energie při běhu ze záložní baterie. Zpočátku se objevil problém, kdy po odpojení zařízení z USB portu byla stále částečně napájena část *LaunchPadu* s USB převodníkem a debuggerem přes *RXD* a *TXD* piny sériového kanálu mikroprocesoru. Za prvé hrozilo poškození mikroprocesoru a za druhé tento odběr v řádu miliampér byl neúnosný při běhu ze záložní baterie. Tento problém se mi podařilo vyřešit změnou funkce *RXD* a *TXD* pinů na digitální vstupy po detekci ztráty napájení z USB. Tímto se celá tato část *LaunchPadu* „odřízne“ od napájení. Po připojení k USB portu se opět obnoví jejich funkce HW UARTu.

Hlavní problém se spotřebou byl zažehnán, avšak stále zařízení spotřebovávalo ze záložní baterie desítky mikroampér. Tento odběr se navíc postupem času stále zvyšoval až ke stovkám mikroampér. Aby se dva vodivostní senzory ve stejném květináči neovlivňovaly připojil jsem napájecí piny *GND* a *VCC* Čínského senzoru oba přes GPIO piny mikroprocesoru. V době, kdy na něm neprobíhá měření, jsou tyto piny nastaveny jako digitální vstupy a pouze když má probíhat měření, je změním na digitální výstupy a jeden na *log 0* a druhý na *log 1*.

Problém spočíval v plovoucím napětí na digitálních vstupech těchto pinů. Jakmile jsem na obou pinech povolil interní *PullDown rezistory*, spotřeba zařízení ze záložní baterie ihned klesla na 2,5  $\mu\text{A}$ . S touto hodnotou jsem byl již spokojen. Takto vznikl kód jehož ukázkou je možné vidět ve výpisu 4. Tento kód se spustí při inicializaci zařízení a poté pokaždé, když se změní stav některého z napájecích zdrojů (USB, či externího).

Výpis 4: Kód pro snížení spotřeby energie

---

```
1 if (P2IN & USB_POWER_INPUT) { // Připojeno k USB
2   P2IES |= USB_POWER_INPUT; // Detekce spádové hrany
3   P1SEL  |= RXD + TXD;      // Nastavit piny pro HW UART komunikaci
4   P1SEL2 |= RXD + TXD;
5 } else {                      // Odpojeno USB
6   P2IES &= ~USB_POWER_INPUT; // Detekce náběžné hrany
7   P1SEL  &= ~(RXD + TXD);    // Nastavit piny jako digitální vstupy
8   P1SEL2 &= ~(RXD + TXD);
9 }
10
11 if (P2IN & EXT_POWER_INPUT) { // Připojeno k externímu zdroji
12   P2IES |= EXT_POWER_INPUT; // Detekce spádové hrany
13   P2REN &= ~(CHINA_SENSOR_GROUND + CHINA_SENSOR_POWER); // Zakázat PullDown
      rezistory
14 } else {
15   P2IES &= ~USB_POWER_INPUT; // Detekce náběžné hrany
16   P2REN |= (CHINA_SENSOR_GROUND + CHINA_SENSOR_POWER); // Zapnout PullDown
      rezistory
17   P1OUT &= ~LED1;           // Pro jistotu, že je indikační LED zhasnutá
18 }
```

---

## 5 Software pro tvorbu křivky

### 5.1 Výběr vývojového prostředí

Pro vývoj uživatelského rozhraní na ovládání mikropočítače jsem zvolil MICROSOFT *Visual Studio* a programovací jazyk C# .NET.

Požadoval jsem tvorbu multiplatformní aplikace (.NET) v objektovém programovacím jazyce s Cěčkovou syntaxí (C#) s dostatečným množstvím užitečných knihoven a jednoduchou (vizuální) tvorbou grafické části aplikace (Visual Studio).

Toto grafické rozhraní jsem začal dělat v rámci své Bakalářské práce [3] a pro použití se závlahovým zařízením jsem ho upravil.

### 5.2 Návrh uživatelského rozhraní

#### 5.2.1 Požadavky

Uživatelské rozhraní by mělo umožňovat tvorbu a úpravy křivky půdní vlhkosti. Mělo by mít možnost křivku přenést do závlahového zařízení. Také by mělo být schopné nakonfigurovat zařízení a nastavit v něm čas. Křivku zobrazenou v programu by mělo být schopné uložit a také otevřít z uloženého souboru.

#### 5.2.2 Vzhled a rozmístění prvků

Program by měl být co nejvíce intuitivní a jeho ovládání by mělo být dostatečně jednoduché, aby umožňovalo pohodlné zadávání hodnot.

#### 5.2.3 Hlavní okno

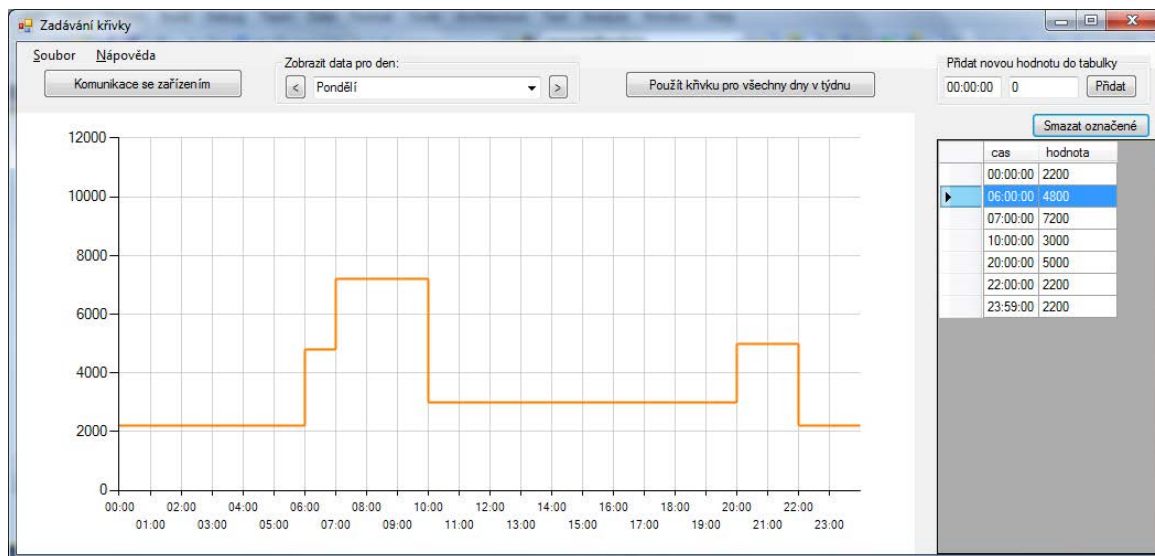
Na *obr. 24* je vidět jak vypadá hlavní okno programu.

Menu *Soubor* umožňuje ukládat/načítat křivku do/ze souboru v počítači. Nad grafem křivky se vybírá den v týdnu, pro který se zobrazuje křivka hodnot. Pod výběrem dne je vyobrazen graf křivky a vpravo od něj jednotlivé hodnoty křivky pro zvolený den.

Při změnách v tabulce hodnot se automaticky překresluje graf křivky. Změny se provádějí dvojklikem na políčko, které chceme změnit. Přidávání nových hodnot se provádí formulářem nad tabulkou a mazání označeného záznamu je uskutečněno po stisku klávesy *delete*, nebo kliknutím na tlačítko *Smazat označené*.

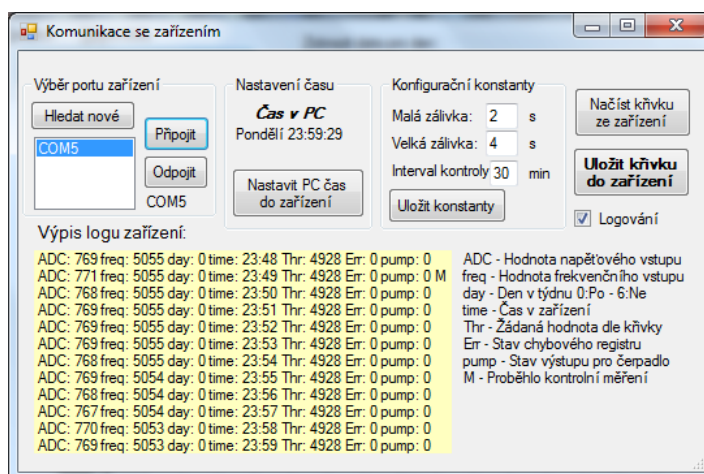
Mezi dny v týdnu se dá přepínat buď změnou dne v padacím menu, nebo pomocí šipek umístěných po stranách padacího menu.

Tlačítko *Komunikace se zařízením* otevře nové okno určené pro komunikaci se závlahovým zařízením.



Obrázek 24: Hlavní okno programu

#### 5.2.4 Okno komunikace se zařízením



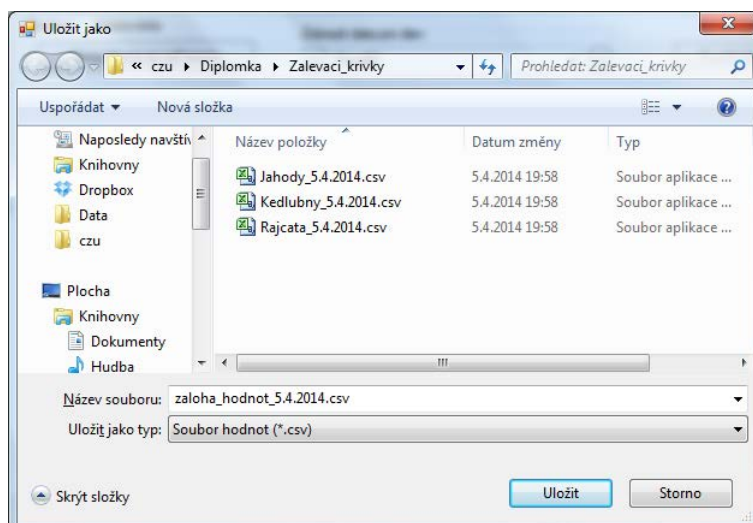
Obrázek 25: Okno Komunikace se zařízením

Na obr. 25 je okno určené pro vlastní komunikaci s mikropočítačem.

Tlačítko *Hledat nové* vyhledá všechny sériové porty v počítači (zařízení se připojuje přes virtuální sériový port) a zobrazí je. Pod tímto tlačítkem je seznam všech vyhledaných portů, kde si uživatel může jeden z nich vybrat. K zařízení se uživatel po vybrání portu může připojit stisknutím tlačítka *Připojit*.

Další políčka mluví samy za sebe. Pomocí sekce *Nastavení času* se dá v zařízení synchronizovat čas s časem nastaveným v PC. V sekci *konfigurační konstanty* může uživatel přizpůsobit regulační konstanty závlahového systému podle použitých komponent a konkrétního nastavení. Poté následují tlačítka pro uložení/přečtení křivky do/ze zařízení a checkbox, kterým se dá spustit logování systému. Systém poté každou minutu odesílá informace o jeho stavu a naměřených veličinách do počítače a záznamy se zobrazují ve žlutém výpisu. Vedle výpisu jsou vysvětlivky, co která zkratka znamená.





Obrázek 26: Dialog pro ukládání souboru

### 5.2.5 Práce se soubory

Pro otevírání a ukládání souborů se díky použití *Windows.Forms* toolkitu používají standardní file-dialogy z Windows viz. obr. 26. Díky možnosti ukládání křivky do souboru ve formátu *csv* je možné uložit si nastavení pro různé druhy rostlin, posílat si je třeba přes internet, nebo si je jen někde zazálohovat. Formát *csv* je běžný standard a je možné ho otevřít a editovat i v jiných programech, například v tabulkových procesorech.

## 5.3 Realizace

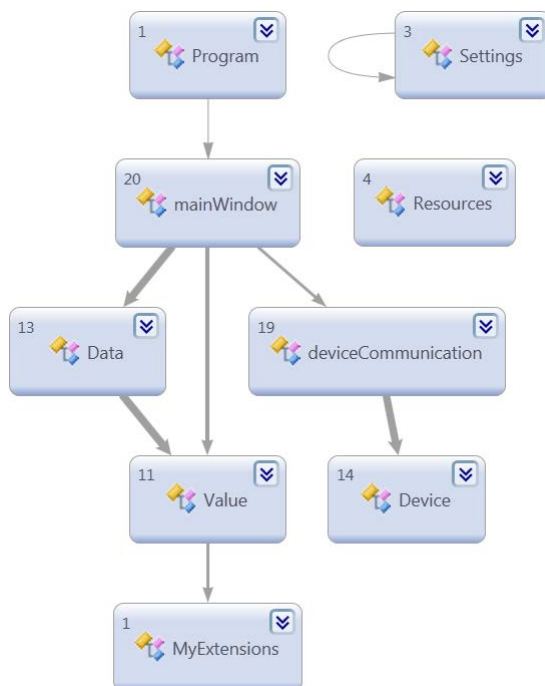
### 5.3.1 Objektový návrh

Na obr. 27 jsou vidět třídy a vztahy mezi nimi. Třída *mainWindow* znázorňuje hlavní okno programu (obr. 24), třída *deviceCommunication* znázorňuje okno komunikace se zařízením (obr. 25). Třída *Data* se má starat o vše spojené s daty, ať už jejich uložení, tak i jejich třídění, či ukládání a načítání ze souboru. Třída *Hodnota* v sobě uchovává čas a hodnotu a je součástí kolekce *values* ve třídě *Data*, kde se všechny hodnoty uchovávají.

Ve skutečnosti se v projektu používá mnohem víc objektů, jako např. *chart* (znázorňující graf), *serialPort* (znázorňující sériový port), či *dataGrid* (zodpovědí za tabulku hodnot), ty jsou však přímo součástí .NET frameworku a v obrázku jsou znázorněny jen třídy mnou definované. Zde je vidět v praxi jak .NET Framework usnadňuje rutinní práci a umožňuje programátorovi se více soustředit na skutečné jádro problému.

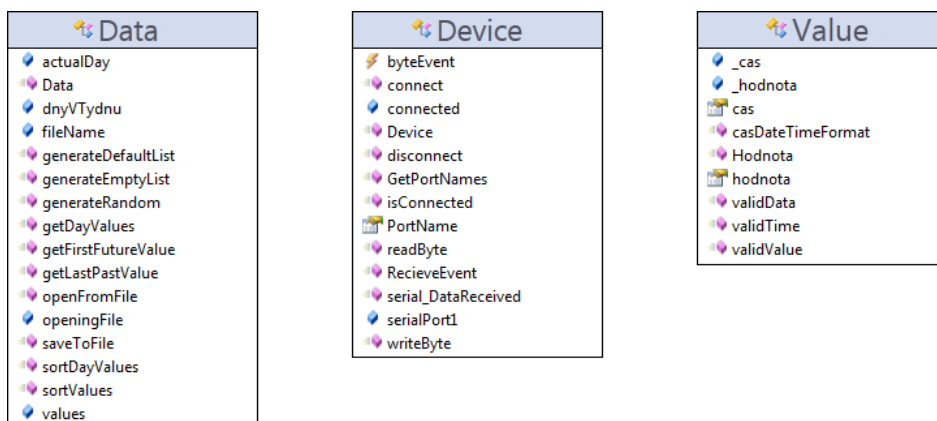
### 5.3.2 Datové operace

O všechnu manipulaci s daty se stará třída *Data*, její struktura je znázorněna na obr. 28. Nejdůležitější je kolekce *values*, v té jsou uloženy všechny hodnoty ze všech dnů. Je to kolekce kolekcí hodnot. Každý



Obrázek 27: Vztahy mezi objekty

den má svoji kolekci hodnot a tyto kolekce jsou sdruženy v kolekci *values*. Pole *dnyVTydn* obsahuje názvy dnů v týdnu pro jejich zobrazení. Metoda *getDayValues* vrací kolekci hodnot pro den zadaný v parametru metody.



Obrázek 28: Struktura tříd Data, Device a Value

Metoda *sortValues* seřadí všechny hodnoty ve všech dnech pomocí metody *sortDayValues*, která s využitím delegování seřadí hodnoty podle času v daném dni. Třídění probíhá po každé editaci, přidání nové hodnoty, či načtení ze souboru. Jednak kvůli přehlednosti, jednak kvůli kontinuitě křivky v grafu. Aby bylo zobrazení grafu kompletní, včetně návaznosti na minulý a následující den, získají se poslední hodnota z předchozího dne a první hodnota ze dne následujícího pomocí metod *getFirstFutureValue* a *getLastPastValue* a tyto se přidají ke grafu, čímž se zajistí kontinuální křivka.

Třída *Value* (obr. 28) reprezentuje jednotlivou hodnotu. Každá instance třídy *Value* obsahuje čas a hodnotu samotnou. Tyto instance jsou pak uloženy do seznamu *Data.values*. Třída *Value* má několik

přetížených konstruktorů, každý pro různý formát času a hodnoty (string, či *TimeSpan* / string, či *Integer*). Pro převod času ze *stringu* do *TimeSpanu* je s výhodou použita technologie *Extension Methods*, která je součástí .NET frameworku od verze 3.5 (C# verze 3.0). Ve třídě *MyExtensions* je nadefinovaná „Extension“ *toTimeSpan(this String str)*, která funguje podobně jako metoda *toString()*, akorát převádí řetězec ve formátu "00:00", či "00:00:00" na typ *TimeSpan*.

Třída *Data* se stará také o ukládání a nahrávání dat ze souboru. Z hlediska čistě objektového by tuto činnost mohla dělat nová třída *File*, avšak třída *Data* toho nemá tolik na starosti aby nemohla obsluhovat i ukládání a načítání souborů, koneckonců je to také operace s daty. Proto k těmto účelům obsahuje třída *Data* metody *saveToFile* a *openFromFile*.

Formát souboru pro ukládání dat jsem zvolil léty osvědčený \*.csv. Je to klasický textový soubor, jehož struktura vypadá následovně:

---

```
1 0;03:00:00;19
2 0;04:00:00;14
3 0;12:00:00;11
```

---

Každý záznam je na novém řádku a jednotlivé položky jsou odděleny středníkem. V našem případě je první hodnota číslo dne v týdnu (0-6), kde číslo 0 znamená pondělí a 6 neděli. Druhá položka je časový údaj a třetí položka je hodnota půdní vlhkosti pro daný den a čas. Formát *csv* je možné otevřít i obyčejným textovým editorem, či tabulkovým procesorem (Microsoft Excel, OpenOffice Calc, atp.), což je výhodné.

Metoda *saveToFile* ze zobrazených hodnot vytvoří tento *csv* soubor a metoda *openFromFile* naopak tento soubor rozparsuje a vytvoří ze získaných dat kolekci ve formátu kolekce *Data.values*.

### 5.3.3 Zobrazení dat

O zobrazení dat se stará třída *mainWindow*. V této třídě jsou definovány všechny obslužné funkce událostí hlavního okna. Například při stisku (přesněji až při uvolnění) klávesy v zadávacím poli *tbHodnota* (pole pro zadávání nové hodnoty) se zkontroluje zda stisknutá klávesa byla klávesa *Enter* a pokud ano, vyvolá se událost kliknutí na tlačítko *Přidat* a také se přesune kurzor do zadávacího pole pro čas, aby bylo možné pohodlně zadat další hodnotu. Tímto je zajištěno jednoduché přidávání hodnot pomocí klávesy *Enter*. Tato jednoduchá obslužná funkce je k vidění na následující ukázce kódu.

Pro zobrazování dat je důležitá metoda *viewDay(int actualDay)*, při jejímž zavolání se aktualizuje tabulka hodnot (*dataGrid*) a zavolá se metoda *populateChart(actualDay)*, která má na starosti překreslení grafu dle aktuálních hodnot ve třídě *Data*. Metoda *viewDay* se volá po každé změně v kolekci *Data.values* (tj. mazání, přidávání, editace záznamu, změna dne, načítání nových hodnot, atd.).

Aktuální den, neboli den, který se zrovna zobrazuje uživateli se získává jako vlastnost *SelectedIndex* objektu *dayCombo*, který reprezentuje *ComboBox* výběru dne.

### 5.3.4 Komunikace se zařízením

Okno *Komunikace se zařízením* reprezentuje třída *deviceCommunication*. Ta krom metod zajišťujících obsluhu grafických prvků obsahuje také metodu *dataReceived*, která je volána ze třídy *Device* (její struktura na *obr. 28*) ve chvíli, kdy na sériový port přijdou data. Přijatá data avšak nejsou kompletní, událost *dataReceived* nastává opakovaně během přenosu a data přicházejí po nedefinovaných částech, je proto potřeba data zkompletovat a rozdělit na přijaté řádky pomocí kterých zařízení komunikuje.

Funkce *dataReceived* je volána z jiného vlákna (threadu) než v jakém vlákně probíhá veškeré zobrazování formulářových prvků. Pro zobrazení dat musíme použít delegace, ta je v C# označovaná jako *Control.Invoke Method*. Tato metoda spustí delegovanou funkci na vlákně, které má kontrolu nad žadaným prvkem. Tak je zajištěna bezpečná mezi-vláknová komunikace. V následující krátké ukázce kódu je znázorněno jak přijímání dat ze sériového portu funguje.

Výpis 5: Přijímání dat přes sériový port

---

```

1 void dataReceived(object sender, ushort value){
2     SerialPort sp = (SerialPort) sender;
3     dataBuffer += sp.ReadExisting();           // doplnit dataBuffer přijatými daty
4     var regex = new Regex(@"(".*\r\n")");     // regexp rozdělení na řádky
5     do{
6         match = regex.Match(dataBuffer);     // kontrola regulárního výrazu
7         if (match.Success){                 // řádek nalezen
8             this.Invoke((MethodInvoker) delegate{ // delegovaná metoda
9                 this.inputLine = match.Captures[0].Value; // uložit řádek
10                refreshUi(); // zpracovat řádek, tento kód se spouští v jiném vlákně
11            });                               // konec delegované metody
12            // vymazat uložený řádek z dataBufferu
13            dataBuffer = dataBuffer.Remove(match.Captures[0].Index,
14                                           match.Captures[0].Length);
15        }
16    } while (match.Success);                 // načítat řádky dokud jsou

```

---

Třída *Device* se stará o komunikaci se zařízením a vše s ní spojené. Jsou to operace, jako připojení k zařízení, přijímání a odesílání dat, detekce chyb atp.

Zařízení odesílá data v ASCII kódu a přijímá data ve vlastním formátu. To je proto, aby se dalo logování zařízení jednoduše použít i s jiným softwarem v případě logování měřených hodnot. Při posílání dat do zařízení se nejdříve pošle příkaz (číslo zabírající 8 bitů v bajtu tzn. větší jak 128)

- 130 - nastavení času, poté následují 4 bajty, každý reprezentuje číslo po sobě jdoucích proměnných Den, Hodina, Minuta, Sekunda, systém odpoví v ASCII kódu potvrzující zprávou sdělující nastavený čas.
- 131 - tabulka hodnot, poté následuje bajt s počtem hodnot v odesílaném bloku (max. 64), dále pořadové číslo bloku a poté již samotné hodnoty bloku rozdělené na spodních 7 bitů a horních 7 bitů jdoucích po sobě. První dva bajty jsou čas a následující dva jsou hodnota vztažená k předešlému času, názorná ukázka ukládání křivky je ve výpisu 6.

- 132 - malá zálivka, následuje spodních 7 bitů, poté horních 7 bitů hodnoty
- 133 - velká zálivka, následuje spodních 7 bitů, poté horních 7 bitů hodnoty
- 134 - interval kontroly , následuje spodních 7 bitů, poté horních 7 bitů hodnoty

---

Výpis 6: Komunikační protokol

---

```

1 // Odeslaný Příkaz
2   131
3 // Odeslaná Data (mezera odděluje bajty)
4 // R B T T V V T T V V
5   4 1 0 1 0 4 0 2 2 1
6 // Přijatá Odpověď po odeslání celého bloku (odesílají se zpět první 3 uložené
   časy s hodnotami pro kontrolu)
7 // Poč Čas Hod Čas Hod
8   4 128 512 256 130

```

---

**R** - Row Count (počet hodnot v bloku), **B** - Block Index (pořadové číslo bloku), **T** - Time (Spodních/Horních 7 bitů času), **V** - Value (Spodních/horních 7 bitů hodnoty), **Poč** - celkový počet doposud uložených hodnot

## 6 Ověření funkčnosti zařízení

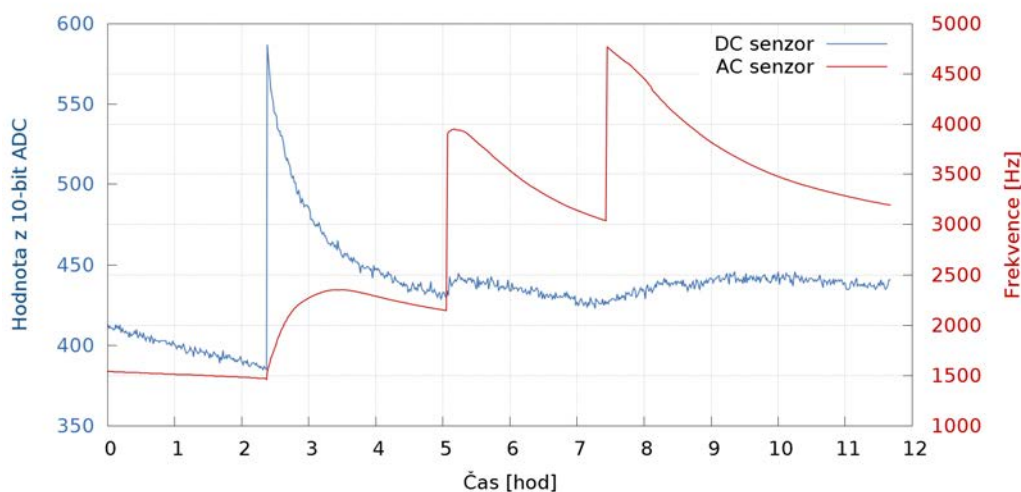
### 6.1 Měření vlhkosti půdy

Dříve než jsem se pustil do regulace, načítání hodnot křivky a podobně, jsem provedl měření vlhkosti v čase na obou senzorech. Na Čínském senzoru z ebaye (6) i na senzoru mé konstrukce (3.2.5). Potřeboval jsem zjistit průběh vlhkosti v čase, reakci senzorů na závlahu a celkovou charakteristiku a kvalitu měření. Podle naměřených dat jsem byl schopen navrhnout vhodný model regulace a regulační konstanty, spolu s intervalem kontrolních měření.

Naměřil jsem velké množství dat. Podělím se o to nejzajímavější formou grafů.

#### 6.1.1 Vliv umístění senzoru

Průběh měřené vlhkosti v čase na čidle závisí na jeho umístění vůči přítoku vody z čerpadla. Umístil jsem oba senzory do stejného vyschlého květináče. Každé k jednomu okraji, tak aby nebyly u sebe. Na obrázku č. 29 je vidět měřený průběh na obou senzorech během postupně jdoucích 3 zalití stejnou dávkou vody (20 ml). První dávka byla naměřena na DC senzor (Čínský), druhá a třetí na AC senzor (mé konstrukce). Z grafu je patrná rapidní změna měřené veličiny na senzoru ke kterému byla přivedena voda, následné klesání vlhkosti zapříčiněné vsakováním vody do okolní zeminy a postupný nárůst vlhkosti na druhém senzoru, u kterého zemina nasává vlhkost. Zároveň je i vidět, že mnohem lepší výsledky podává senzor využívající střídavého proudu k měření. Měřítka pro oba senzory jsem získal výpočtem vycházejícím z maximální měřené hodnoty. Provedl jsem i test ve sklenici s vodou.

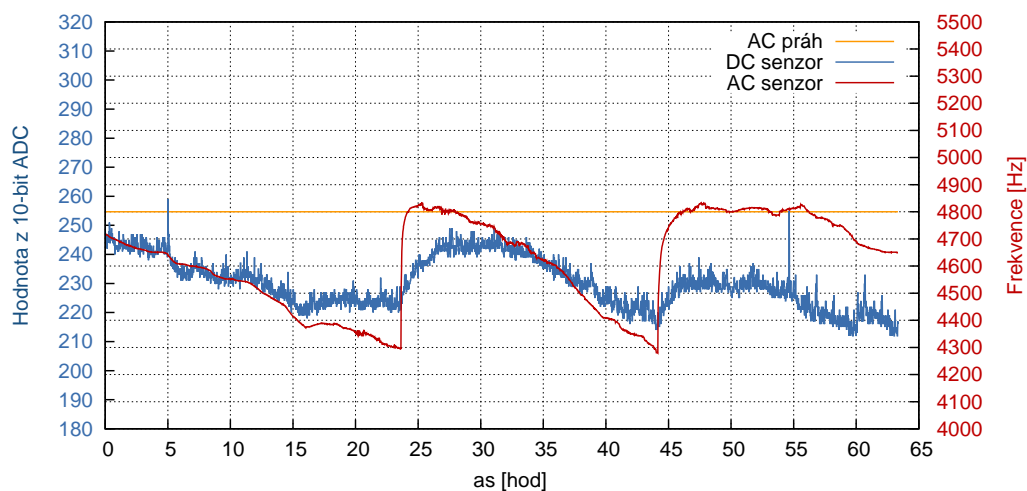


Obrázek 29: Vliv umístění přítoku vody vůči senzoru na průběh měření

#### 6.1.2 Kontrola kvality měření a regulace

Sestrojil jsem regulaci podle hodnot AC senzoru a nastavil jsem zařízení, aby udržovalo hodnotu 4800 Hz, na grafu z obr. č. 30 je to žlutá čára pojmenovaná AC práh. Regulaci jsem nastavil, aby

se provedla dvousekundová záливka v případě poklesu větším než 500 Hz pod tento práh. Výsledek měření mě velmi mile překvapil, měřená hodnota oscilovala mezi 4800 Hz a 4300 Hz s poměrně dosti podobnými křivkami vysychání a zalévání. Testovací květináč jsem položil na digitální kuchyňskou váhu a zkoumal hmotnost těsně před zalitím a po zalití. Velikost záливky se v průběhu dvoutýdenního pozorování pohybovala v rozmezí od 19 g do 22 g. Hmotnost květináče po záливce se lišila maximálně o 4 g. To samo o sobě hodnotím jako úspěch. V grafu pořízeném během tohoto pozorování jsem vynesl i data z DC senzoru. Zalévání probíhalo z volného konce hadičky doprostřed mezi oba senzory. Křivky obou senzorů jsou vůči sobě v měřítku a jak je vidět i když se částečně překrývají, DC senzor podává značně zkreslené údaje. Navíc je i velmi roztřepená a to je přitom každý vzorek počítán průměrem z osmi naměřených hodnot.



Obrázek 30: Porovnání kvality měření obou senzorů

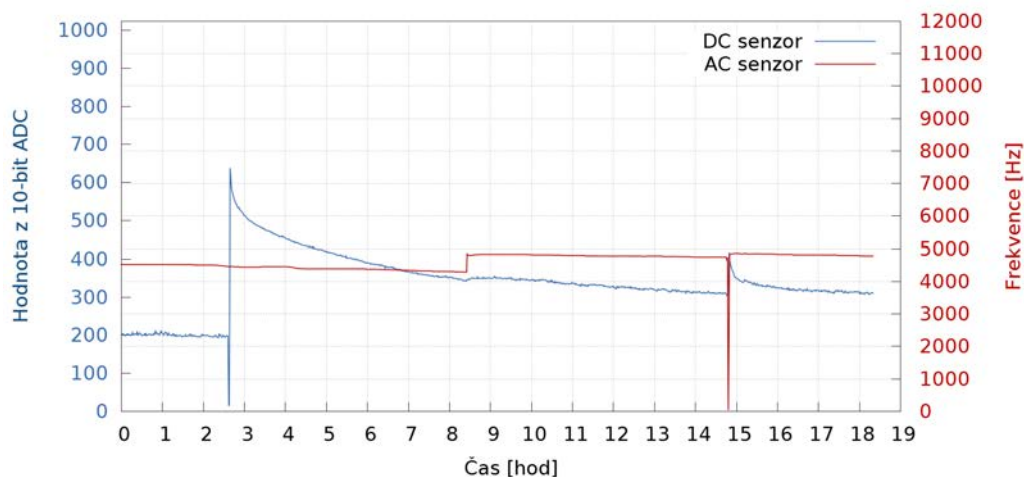
### 6.1.3 Test stability měřených veličin

Zajímalo mě také, jak se dá na měřené hodnoty spolehnout. Provedl jsem sérii testů, z nichž ten nejzajímavější je možno vidět na obrázku č. 31. Nejprve jsem vytáhl z půdy DC senzor (okolo času 2:30) a vzápětí ho zasunul na téměř totožné místo. Výsledek je nejprve pokles měřené hodnoty téměř k nule, což je v pořádku, avšak co následuje po vrácení senzoru do zeminy je vskutku pozoruhodné. Rozsahy obou ypsilonových os jsou záměrně nastaveny tak, aby pokrývaly plné měřitelné rozsahy obou senzorů. To proto, aby bylo vidět, že rozdíl měřené hodnoty DC senzoru po vrácení do půdy je více než o 30 % měřeného rozsahu vyšší. Tudíž je tento senzor skutečně nevhodný.

AC senzor mé konstrukce naopak vykazuje velmi dobré výsledky v tomto testu. Provedl jsem s ním to samé co s DC senzorem a jak zareagoval je možné vidět před začátkem 15-té hodiny. Jeho změna

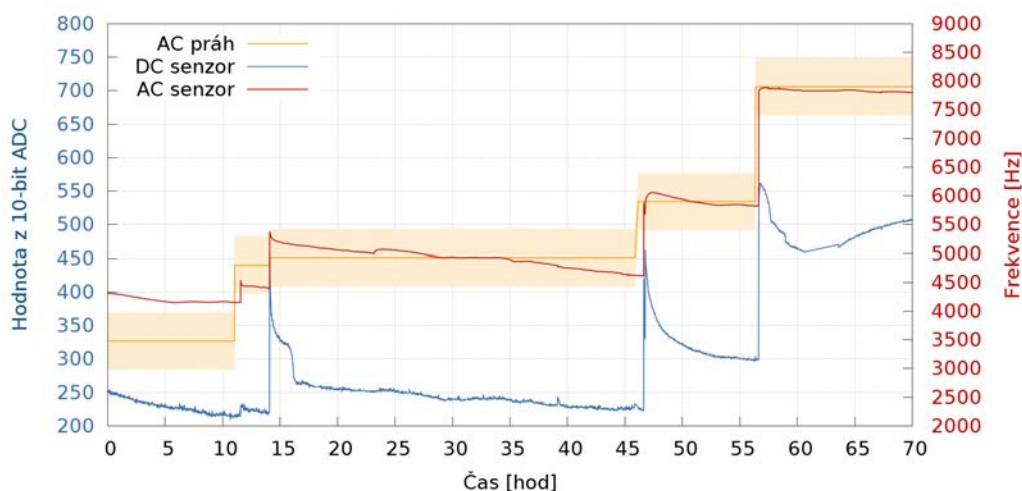
měřené veličiny po znovu vložení senzoru do půdy byla okolo 1 % z rozsahu. To já osobně považuji za velký úspěch.

Na stejném grafu je také přibližně uprostřed vidět jedno zalití tzv. „malá zálivka“ (po 8-mé hodině). Jednak je vidět, jak jemná je změna způsobená malým zalitím. Také jak jak rychlá a přesná může být odezva při správném namíření hadičky do květináče vůči senzoru. A v neposlední řadě, jak si toho zalití Čínský senzor téměř nevšiml.



Obrázek 31: Test stability měřených veličin

## 6.2 Regulace podle křivky



Obrázek 32: Průběh půdní vlhkosti během regulace podle křivky

Ověřil jsem, že AC senzor je možné použít pro regulaci podle křivky. Přistoupil jsem tedy k doprogramování podpory pro křivku a zbývalo jediné - ověřit funkci systému v praxi. Poté co jsem naprogramoval podporu pro sledování aktuální hodnoty vlhkosti půdy podle zadané křivky (tabulky) jsem



uložil do zařízení jednoduchou křivku s 5-ti hodnotami {3478; 4800; 4928; 5900; 7900} pro různé časy. Následně jsem spustil logování hodnot a vznikl graf na obrázku č. 32. Pro zajímavost jsem do grafu vynesl i hodnoty z Čínského senzoru. Dosáhnout s tímto senzorem fungující regulace by bylo prakticky nemožné. Avšak regulace podle křivky funguje velmi dobře s použitím senzoru mé konstrukce.

První tři zalití jsou tzv. „malá zálivka“ až v přechodu mezi hodnotami 5900 a 7900 je rozdíl mezi měřenou a žádanou hodnotou větší a nastala „velká zálivka“. Jak je vidět v obou případech se skutečná hodnota drží v regulačním pásmu šíře 1 kHz vyznačeném světle oranžovým podbarvením. Žádaná hodnota je znázorněna oranžovou linkou. Z grafu je patrný i interval kontrolních měření, který jsem nastavil na 30 min. Projevuje se zpožděním zálivky za vzestupem žádané hodnoty. Další zajímavý moment je hned první zalití, hodnota vlhkosti nestoupá o tolik jako při následných zalitích. To bylo způsobeno prázdnou hadičkou po instalaci zdokonalené zavlažovací koncovky.

### 6.3 Vyhodnocení

Systém splnil požadavky zadané na začátku. Je schopný zalévat a udržovat hodnoty vlhkosti zadané křivkou. Senzor vlastní konstrukce podává dobré výsledky, rychlou reakční dobu, stabilitu měřených hodnot, avšak měřené hodnoty jsou závislé na slanosti substrátu. Pro konstrukci jednoduchého závlahového automatu vyhovuje za předpokladu, že není potřeba obecné kalibrace na veličiny vlhkosti půdy. Ani po třech měsících provozu neztratily nerezové elektrody svůj lesk a nejeví známky oxidace. Čínský senzor prodávaný na aukčním serveru Ebay.com naopak zoxidoval již po pár týdnech používání, navíc výsledky jeho měření jsou velice nestabilní a nepřesné. Vyhodnotil jsem tento senzor jako naprosto nevhodný pro toto použití.



Obrázek 33: Zavlažovací koncovka z provrtané hadice

Rychlost reakce senzoru na provedenou zálivku závisí na způsobu zavedení vody do měřené zeminy. Skvělé výsledky v květináči jsem zaznamenal s perforovanou hadičkou stočenou dokola po okraji květináče (viz. obr. č. 33). Jedná se o obyčejnou PVC hadičku s navrtanými dírkami. Díky tomuto uspořádání je možné do několika desítek minut změřit výslednou hodnotu, která se již příliš nemění, naopak při pouhém zavedení hadičky na jedno místo v květináči trvá i několik desítek hodin, než se voda vsákne rovnoměrně. Toto uspořádání bylo použito i při finálním měření pro ověření funkce zařízení (graf viz. obr. č. 32).

## 7 Závěr

Cílem práce bylo navrhnout a sestavit levné zařízení schopné zalévat půdu podle křivky půdní vlhkosti. Cíl práce byl splněn. Zařízení jsem sestavil a funguje podle očekávání. Lze jej připojit k počítači, uložit do něj křivku půdní vlhkosti a podle ní pak zařízení zavlažuje zeminu. Zařízení dokáže detekovat poruchu senzoru, čerpadla, poškozené vedení vody k záhonu/květníku, či absenci vody v nádrži. O detekované poruše dá zařízení vědět chybovým blikáním. Zařízení dokáže také rozpoznat poškozená, či chybějící data potřebná pro regulaci půdní vlhkosti. Zařízení je vybavené vstupem pro senzor vlastní konstrukce (viz. 3.2.5), který podává dostačující výsledky pro domácí použití, avšak k jeho měřené hodnotě neexistuje kalibrace na jednotky objemové vlhkosti půdy. Zařízení je vybavené i druhým vstupem, ke kterému je možné připojit senzor s napěťovým výstupem (0-3 V). Při použití senzoru ke kterému existuje kalibrace (např. VH400 viz. 3.2.3, nebo senzor VIRRIB viz. 3.2.4) je možné měřit i v jednotkách objemové vlhkosti. Zařízení je také vybaveno dvěma výstupy, ke kterým je možné připojit čerpadlo, či elektromagnetický ventil pro ovládání tlakové vody.

Při stavbě zařízení jsem použil levné a recyklované součástky. Nejdražší na celém zařízení je LaunchPad, který od doby psaní mé Bakalářské práce podražil na necelých 10 USD. Pořizovací náklady celé řídicí jednotky se tak vejdu do 400 Kč i s LaunchPadem, který jsem použil pro stavbu prototypu z důvodu snadného ladění vznikajícího kódu. Pokud bych chtěl toto zařízení vyrábět komerčně, použil bych místo LaunchPadu mikroprocesor, který je vybaven USB portem (viz. tabulka 2). Cena těchto mikroprocesorů bývá do 2 USD tudíž výsledná cena takového zařízení může být ještě podstatně nižší. Čerpadlo jsem použil z vyřazeného automobilu a senzor jsem vyrobil z odpadního materiálu.

Návrhů na vylepšení mě napadá mnoho. Například rozšíření kalendáře na celý rok, jelikož některé druhy rostlin, vyžadují různé podmínky v různých fázích růstu. Já zvolil pouze týdenní cyklus především proto, abych mohl ověřit jeho funkčnost a svěřím se, že odlaďování, byť jen týdenního jevu je velmi časově náročné. Použitý mikroprocesor také obsahuje teplotní senzor, dalším zlepšením by mohlo být využití tohoto senzoru při regulaci závlahy. Ještě lepší by bylo doplnit zařízení senzorem půdní teploty. Případně by se dal systém rozšířit na regulaci více veličin ovlivňujících růst rostlin, jako je teplota (ventilace), vzdušná vlhkost apod.. Kombinace systému s dešťovým senzorem by mohla ještě zlepšit chování systému ve venkovním prostředí. Ten by tak dokázal předvídat navýšení půdní vlhkosti ještě před tím, než zemina nasákne srážkovou vodou. V době chytrých telefonů by bylo určité zajímavé a mnohými žádané mít možnost ovládat zařízení přímo pomocí chytrého telefonu.

Na loňském veletrhu FOR GARDEN 2013 získal cenu Grand Prix senzor vlhkosti půdy *Precision<sup>TM</sup> soil sensor* od firmy TORO . Tento senzor sleduje procentuální úroveň nasycení půdního profilu vodou a podle fixně nastavené úrovně vlhkosti dokáže blokovat naplánovanou závlahu. Kromě této funkce je schopen sledovat ještě teplotu půdy a vypnout závlahu při nízkých teplotách . Tento senzor byl oceněn pro své unikátní vlastnosti a zejména inovativní technologii pomáhající šetřit vodu. Maloobchodní cena tohoto senzoru je 3600 Kč, ten je přitom pouze doplňkovým zařízením k řídicí jednotce. Domnívám se, že oproti mému zařízení vyniká tento výrobek především dobrým designovým zpracováním. Po technické stránce ovšem mohu nabídnout dokonalejší řešení řídicího procesu díky počítačovému nastavení závlahové křivky v čase a nezanedbatelnou výhodou mého výrobku je i několikanásobně nižší pořizovací cena. Ta by byla udržitelná i v případě sériové výroby a distribuce do obchodní sítě.

## Reference

- [1] Best DIY cheap soil moisture sensor. 2010.  
URL <<http://www.cheapvegetablegardener.com/how-to-make-cheap-soil-moisture-sensor-2/>>
- [2] Senzor vlhkosti půdy - 6440. Březen 2014.  
URL <<http://meteostanice.e-pocasi.cz/1123-senzor-vlhkosti-pudy---6440.html>>
- [3] Bůžek, M.: Uživatelské rozhraní pro ovládání mikropočítačového systému pomocí PC. 2012.
- [4] Bittelli, M.: Innovative methods for measuring soil water content. 2003.  
URL <[http://www.distagenomics.unibo.it/wuemed/M\\_bittelli\\_innovative\\_methods.pdf](http://www.distagenomics.unibo.it/wuemed/M_bittelli_innovative_methods.pdf)>
- [5] Davies, J.: *MSP430 Microcontroller Basics*. Elsevier Science, 2008, ISBN 9780080951577.  
URL <<http://books.google.cz/books?id=qVjhtNYvGGAC>>
- [6] DrX: PCB as a Capacitive Soil Moisture Sensor. Listopad 2011.  
URL <<http://zerocharactersleft.blogspot.cz/2011/11/pcb-as-capacitive-soil-moisture-sensor.html>>
- [7] EME\_Systems: Conductivity with the BS2/OWL2. 2002.  
URL <<http://www.emesystems.com/0L2mhos.htm>>
- [8] HOLTEK: HT75XX-1 Datasheet. Prosinec 2006.  
URL <<http://www.gme.cz/img/cache/doc/330/201/ht7533-1-to92-datasheet-1.pdf>>
- [9] Kramer, P.: *Water relations of plants*. Academic Press, 1983, ISBN 9780124250406.
- [10] Kutílek, M.: *Vodohospodářská pedologie*. Praha: Státní nakladatelství technické literatury, 1966.
- [11] Quiring, K.: Starting a USB Design Using MSP430™ MCUs. Září 2013.  
URL <<http://www.ti.com/lit/an/sl4a457/sl4a457.pdf>>
- [12] Rbrainard: Opamp Non-Inverting Test. Červenec 2012.  
URL <<https://www.circuitlab.com/editor/#?id=6d8ma2>>
- [13] Salaš, P.: Rostliny v podmínkách měnícího se klimatu. *Úroda, vědecká příloha*, Říjen 2011, ISSN 0139-6013.
- [14] Socher, G.: Remote flower watering and monitoring. Září 2009.  
URL <<http://tuxgraphics.org/electronics/200908/eth-flower-watering.shtml#01findex2>>
- [15] Szczys, M.: Launchpad not limited to value line chips. Září 2010.  
URL <<http://hackaday.com/2010/09/28/launchpad-not-limited-to-value-line-chips/>>
- [16] TI: *MSP430x2xx Family User's Guide (Rev. I)*. Leden 2012.  
URL <<http://www.ti.com/litv/pdf/slau144i>>

- [17] Tiaz, L.; Ziger, E.: *Plant Physiology - Third Edition*. Sinauer Associates Inc, 2002, ISBN 9780878938230.
- [18] Tůma, J.: *Zavlažujeme zahradu: moderní hospodaření s vodou*. Profi & hobby, Grada, 2001, ISBN 9788024700830.
- [19] Vegetronix: Vegetronix Soil Moisture Sensor Probes FAQ. Březen 2014.  
URL <<http://www.vegetronix.com/Products/VG400/Soil-Moisture-Probe-FAQ.phtml>>
- [20] Vishwanath: Using flash memory in MSP430. Leden 2013.  
URL <<http://cyroforge.wordpress.com/2013/01/01/using-flash-memory-in-msp430/>>

## Seznam použitých zkratek

CMOS	Complementary Metal–Oxide–Semiconductor - technologie výroby integrovaných obvodů. Využívá tranzistorů technologie MOSFET k realizaci logických funkcí. Velmi často je využívána k výrobě mikroprocesorů.
DPS	Deska Plošného Spoje, nebo také Tištěný Spoj. - destička na které jsou elektronické součástky a měděné cestičky zajišťující jejich propojení.
Eagle	Software pro vývoj DPS.
Firmware	Označení pro software, který slouží pro řízení nějakého vestavěného systému (embedded system).
FRAM	Ferroelectric Random Access Memory - nový typ paměti rychlý jako SRAM, avšak s vlastností uchování dat i po odpojení napájení, TI je používá v některých svých MCU místo paměti FLASH
FTDI	Future Technology Devices International Ltd. - Firma zaměřující se především na výrobu integrovaných obvodů pro komunikaci přes USB.
GPIO	General Purpose Input/Output - pin na mikroprocesoru, jehož funkci je možné programově zvolit. Dá se nastavit jako digitální vstup, analogový vstup, digitální výstup a případně i další.
idle	Režim kdy je zařízení v pohotovosti avšak jinak nevykonává žádnou práci. Často se vhodným využíváním idle režimu pomáhá úspoře energie.
Launchpad	Vývojový kit určený pro programování a debugging mikroprocesorů řady MSP430 ValueLine od Texas Instruments. Jedná se o jakousi konkurenci známého Arduina avšak na jiné 16-bit platformě. S cenou do 10-ti dolarů je velmi zajímavý.
LPM3	Low Power Mode 3 - jeden z úsporných režimů mikroprocesorů MSP430. V tomto režimu běží pouze externí oscilátor (typ. hodinový krystal) a je napájena paměť RAM.
MIPS	Million Instructions Per Second - milion instrukcí za sekundu - měrná jednotka rychlosti mikrokontrolérů
SMD	Technologie povrchové montáže elektronických součástek bez nutnosti vrtat díry pro nožičky. Součástky jsou také mnohem menší než klasické (existují různé velikostní řady)
SPI	Serial Peripheral Interface - sériové periferní rozhraní - Používá se pro komunikaci mezi řídicími mikroprocesory a ostatními integrovanými obvody (EEPROM, A/D převodníky, displeje. . .)
Spy-Bi-Wire	Jedná se o serializovaný JTAG protokol vyvinutý firmou Texas Instruments a používaný pro programování mikroprocesorů řady MSP430. Umožňuje naprogramovat mikroprocesor pomocí dvou vodičů RST a TEST.
UART	Universal Asynchronous Receiver Transmitter - zařízení pro asynchronní komunikaci po sériovém portu například počítače

USB Universal Serial Bus - v současné době standardní rozhraní pro komunikaci periferií s PC

USD Americký dolar

watchdog Speciální časovač mikroprocesoru, který zařizuje automatické restartování mikroprocesoru v případě, že zatuhl.

## Seznam obrázků

1	Struktura zeminy[17] . . . . .	3
2	Půdní hydrolimity . . . . .	3
3	Závislost množství vody přístupné rostlinám na typu zeminy a vlhkosti půdy[9] . . . . .	4
4	Průběh kritických hodnot VVK během vegetačního období pro různé plodiny . . . . .	5
5	LaunchPad . . . . .	9
6	Čínský senzor z Ebaye . . . . .	10
7	Watermark 6440 (Gypsum block) . . . . .	11
8	Vegetronix VH400 . . . . .	11
9	Snímač VIRRIB . . . . .	12
10	Senzor vlastní konstrukce bez izolace . . . . .	13
11	Kapacitní senzor podle DrX . . . . .	13
12	Zapojení obvodu pro kapacitní senzor[12] . . . . .	14
13	Čerpadlo ostřikovačů LADA s bazalkou . . . . .	14
14	Trojitý Elmag. ventil z pračky . . . . .	15
15	Schéma řídicí jednotky . . . . .	16
16	Navýšení výstupního napětí stabilizátoru diodou[8] . . . . .	17
17	FTDI modul . . . . .	19
18	Detail DPS . . . . .	20
19	Pohled na vrchní desku zařízení . . . . .	20
20	Vývojový diagram hlavní programové smyčky . . . . .	21
21	Vývojový diagram přerušování časovače 1 a 2 . . . . .	23
22	Vývojový diagram přerušování od vstupních pinů a A/D převodníku . . . . .	24
23	Vývojový diagram přerušování UART . . . . .	25
24	Hlavní okno programu . . . . .	31

25	Okno Komunikace se zařízením . . . . .	31
26	Dialog pro ukládání souboru . . . . .	32
27	Vztahy mezi objekty . . . . .	33
28	Struktura tříd Data, Device a Value . . . . .	33
29	Vliv umístění přítoku vody vůči senzoru na průběh měření . . . . .	37
30	Porovnání kvality měření obou senzorů . . . . .	38
31	Test stability měřených veličin . . . . .	39
32	Průběh půdní vlhkosti během regulace podle křivky . . . . .	39
33	Zavlažovací koncovka z provrtané hadice . . . . .	40



## Obsah příloženého CD

Příložené CD obsahuje následující adresáře:

- **Text** - obsahuje tuto práci ve formátu PDF
- **Datasheets** - obsahuje datasheety a manuály používané při programování
- **PC\_Sources** - obsahuje zdrojové kódy počítačové aplikace v C# jako *Visual Studio* projekt
- **MSP430\_Sources** - obsahuje zdrojové kódy pro mikrokontrolér v jazyce C jako *Code Composer Studio* (Eclipse) projekt
- **Installation\_Files** - soubory potřebné k instalaci programu - tj. návod na instalaci (soubor README), zkompileovaný program a instalační soubor .NET frameworku