

# **STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST**

Obor SOČ: 10. Elektrotechnika, elektronika a telekomunikace

Automatický zavlažovací systém

Kočica Filip

**Kraj: Zlínský kraj**

**Uherský Brod 2016**

# **STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST**

Obor SOČ: 10. Elektrotechnika, elektronika a telekomunikace

## **Automatický zavlažovací systém**

## **Automatic irrigation system**

Autor: **Filip Kočica**

Škola: **Střední průmyslová škola a Obchodní Akademie  
Uherský Brod**

Adresa školy: **Nivnická 1781, 688 01 Uherský Brod**

Kraj: **Zlínský kraj**

Konzultant: **PaedDr. Antonín Mahdal**

**Uherský Brod 2016**

## Prohlášení

*Prohlašuji, že jsem svou práci SOČ vypracoval samostatně a použil jsem pouze podklady (literaturu, SW atd.) uvedené v seznamu vloženém v práci SOČ.*

*Prohlašuji, že tištěná a elektronická verze soutěžní práce SOČ jsou shodné.*

*Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.*

*V ..... dne .....*

*podpis: .....*

## **Poděkování**

Děkuji panu učiteli Antonínu Mahdalovi za obětavou pomoc a podnětné připomínky, které mi během práce poskytoval.

# Anotace

Tato práce pojednává o automatickém zavlažovacím systému, o porovnání reálného komerčního systému s touto prací, zejména pak o řídicím systému a programu, možnosti dálkového přístupu a o fyzické realizaci. Cílem práce bylo zkonstruovat řídicí jednotku i se simulací běhu zavlažovacího systému. Jednotka poskytuje množství různých funkcí a je možno ji ovládat jak přímo z místa umístění modulu, tak i na dálku přes ethernetovou přípojku.

**Klíčová slova:** Automatický zavlažovací systém; řídicí jednotka Arduino Mega 2560; webové rozhraní; program; LCD; systém

# Annotation

This thesis discusses an automatic irrigation system by comparing the actual commercial system with this thesis, especially the control system and program options for remote access and physical implementation. The main aim was to construct a control unit and a simulation run the irrigation system. The unit provides a variety of functions and can be controlled both directly from the location of the module as well as remotely via the Ethernet connection.

**Keywords:** automatic irrigation system; Controller Arduino Mega 2560; Web interface; program; LCD; system

# Obsah

Úvod .....	7
1 Stanovení cíle .....	8
K hlavnímu cíli jsem si stanovil další cíle: .....	8
2 Požadované parametry a vlastnosti systému .....	10
Požadované parametry: .....	10
3 Automatické zavlažovací systémy .....	11
Popis automatického zavlažovacího systému .....	11
Účel a uplatnění v průmyslu .....	11
Instalace a obsluha .....	11
4 Softwarová výbava .....	13
Ethernetové rozhraní .....	13
Funkce stránky .....	13
Průběh nastavování .....	14
Vzhled jednotlivých záložek na Ethernetu doplněn o vzhled této funkce na LCD .....	15
Hlavní menu .....	15
Záložka dny (Po-Ne) .....	16
Záložka Nastavení parametrů .....	17
Záložka Režim + Kalibrace .....	18
Záložka Vstupy .....	19
Záložka Výstupy .....	20
Záložka Aktuální čas .....	21
Řídící jednotka a program .....	22
Možnosti řízení systému .....	22
Rozložení dat na displeji .....	23
Základní struktura programu .....	24
5 Fyzická realizace .....	25
Popis vytvořené simulace zavlažování .....	26
Blokové schéma .....	26
Schéma zapojení .....	27
Plošný spoj .....	28
Osazovací plán .....	28

6	Technický koutek.....	29
	Třída Zapouzdření dat.....	29
	Funkce loop.....	30
	Procedura zápis do EEPROM.....	31
	Princip zpracování dat načtených z ethernetu do vyrovnávací paměti .....	32
7	Stručný výčet všech funkcí systému .....	33
8	Porovnání této práce s reálným komerčním systémem nabízeným na trhu .....	34
	Závěr.....	35
	Použitá literatura a zdroje.....	36
	Seznam obrázků .....	37

# Úvod

Zavlažovací systémy se v současné době stávají již samozřejmou součástí většiny okrasných zahrad a trávníků. S tím jak se rozšiřují a dostávají do povědomí široké veřejnosti, se stále zvyšují požadavky ze strany zákazníků, a to především v oblasti automatizace celého systému. Podle mého průzkumu v prodejnách těchto systémů se zákazníci stále častěji orientují na systémy, které splňují nejen základní požadavky, ale také je chtějí ovládat vzdáleně.

Smyslem této práce je vytvořit funkční model zavlažovacího systému, který splňuje kritéria zákazníků nejen v oblasti automatizace, ale také vzdáleného přístupu, díky kterému je možné bez fyzické přítomnosti měnit nastavení celého systému. Výhoda navrhovaného řešení spočívá v tom, že celý řídicí program vznikl na základě požadavků ze strany uživatelů zavlažovacích systémů, a je tak přizpůsoben aktuálním požadavkům a zároveň lokalizován v českém jazyce pro snadnou obsluhu systému. Díky využití jednotky Arduino je možné řídicí program měnit i v případě nestandardních požadavků ze strany zákazníků prostřednictvím USB kabelu.

Vytvořený model je zcela funkční a plně vystihuje veškeré situace, které mohou v oblasti zavlažování trávníků nastat.

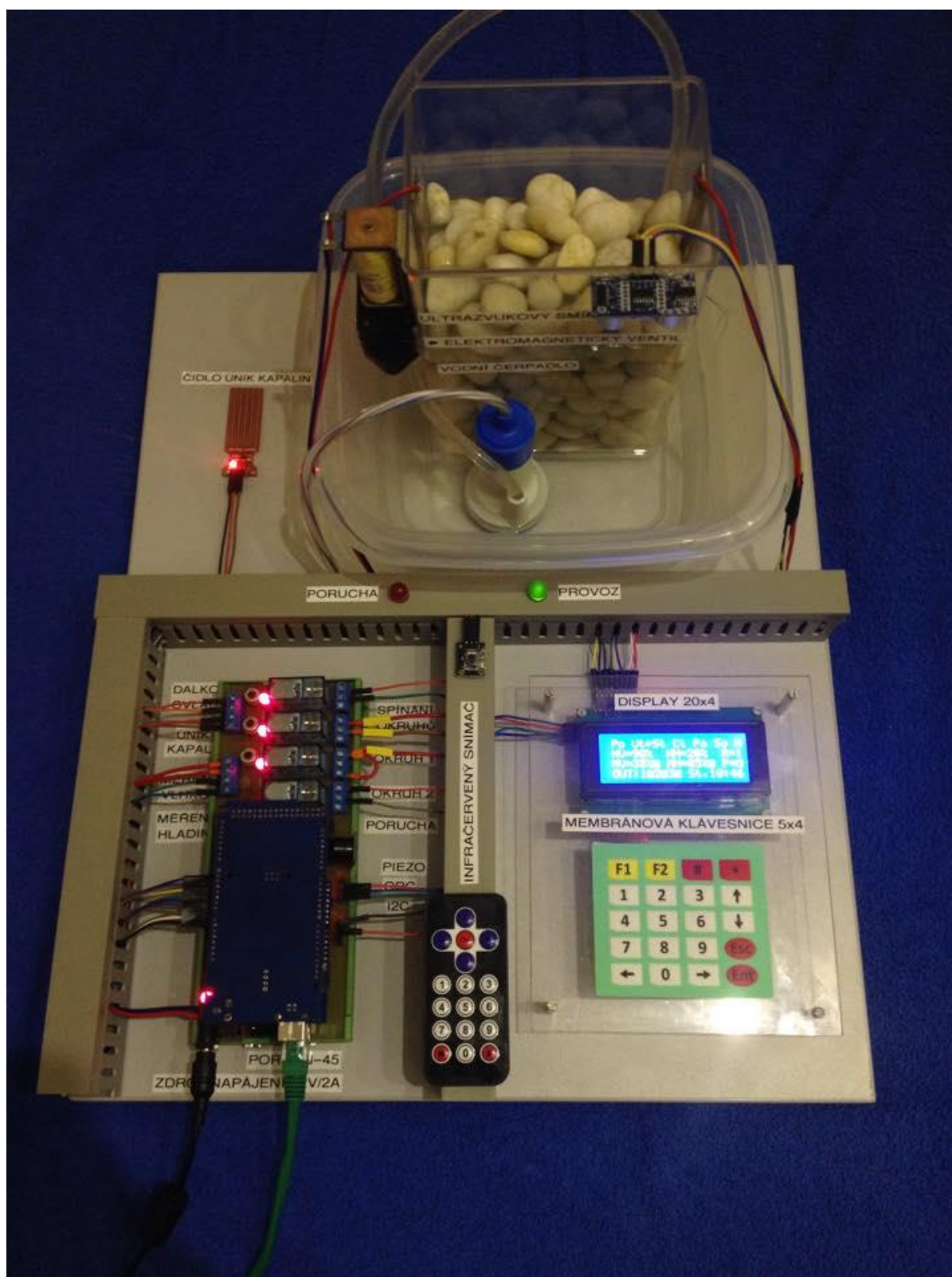


# 1 Stanovení cíle

Na začátku tvorby systému jsem si stanovil několik cílů, které vyplynuly z požadavků na rychlé a pohodlné nastavování a získávání informací ze systému. Hlavním stanoveným cílem tedy bylo vytvořit funkční model zavlažovacího systému a systém, který bude poskytovat uživatelům zásadní informace o aktuálním stavu systému a snímaném prostředí.

## **K hlavnímu cíli jsem si stanovil další cíle:**

- Vytvoření dálkové správy systému, kde bude možno provádět stejné operace jako z lokální stanice
- Možnost nastavení všech potřebných parametrů, které zavlažovací systém ke své správné činnosti potřebuje, ale také i těch zanedbatelných, které nejsou pro chod systému stěžejní, ale spíš doplňkové, a také možnost manuálního spouštění výstupů
- Čtení vstupů, přepočítání na procenta podle kalibrované hodnoty a výpis na displej i ethernet, snímání a následné hlášení poruchy a možnost manuální blokace systému



Obrázek 1: Fotografie výsledného výrobku

## 2 Požadované parametry a vlastnosti systému

Požadavkem bylo zprovoznění řídicí jednotky a vytvoření modelu se simulací plně automatizovaného zavlažovacího systému, který by byl schopen konkurovat obdobným komerčním řešením poskytovaným na trhu a to s co nejnižšími náklady na výrobu.

### Požadované parametry:

- Možnost na řídicí jednotce nastavovat všechny potřebné i doplňkové parametry.
- Dokázat s maximální přesností snímat pomocí analogových vstupů všechny měřené veličiny.
- Vytvořit poruchu a pozastavení poruchy, hlášení pomocí piezo-bzučáku.
- Pomocí DC 12V/2A čerpadla a elektromagnetického ventilu načerpat vodu z nádrže do zavlažovaného prostředí a pak zajistit, aby se voda s časovou prodlevou dokázala sama vylít zpět do nádrže a tím simulovat vypařování vody ze zavlažovaného prostředí a fyzické doplnění vody do nádrže.
- Možnost vzdáleného přístupu se všemi možnostmi, kterými disponuje lokální stanice.
- Navržení a vytvoření plošného spoje (PLC) pro snadnější realizaci a větší přehlednost.
- Náklady do 3000 Kč.

## 3 Automatické zavlažovací systémy

V této kapitole naleznete základní informace o tom, co automatický zavlažovací systém je, jaké je jeho uplatnění v průmyslu, a porovnání funkcí komerčního zavlažovacího systému s touto prací.

### Popis automatického zavlažovacího systému

Automatický zavlažovací systém bývá zpravidla složen z řídicí jednotky, elektroventilů, snímačů, zásobníku vody (zásobárna dešťové vody, studna, vodovodní řád), podzemního trubního vedení, na kterém jsou připojeny výsuvné postřikovače, které pokrývají celou plochu zavlažovaného prostředí. Automatický znamená, že ke svému chodu vůbec nepotřebuje lidský zásah, pouze počáteční nastavení, popřípadě opravu poruchy. Systém je také možno doplnit o různá čidla podnebních podmínek, aby tak řídicí jednotka mohla přizpůsobit program zavlažování aktuálnímu počasí.

### Účel a uplatnění v průmyslu

Automatické zavlažovací systémy se v současné době stávají již samozřejmou součástí ne jen soukromých okrasných zahrad a trávníků, ale i komerčních prostranství, jako jsou golfové hřiště, fotbalová hřiště, parky apod.

Za stěžejní účel a největší výhodu lze považovat ušetření velkého množství času a práce, které by muselo být investováno pro pěkný trávník a zahradu. Další, ačkoli na první pohled ne úplně zřejmou, výhodou je velmi kvalitní závlaha. V neposlední řadě je také systém několika postřikovačů estetičtější, než vodovodní hadice roztahané přes celou zahradu.

### Instalace a obsluha

V dnešní době stačí kontaktovat příslušnou firmu, která potřebuje vědět pouze to, jaký zdroj vody bude systém používat (zásobárnu dešťové vody, studnu, vodovodní řád), typ čerpadla, rozměry a terén zavlažované plochy. Firma sama udělá návrh. Pokud se zákazníkovi bude líbit, pak ho může zrealizovat zcela bez zásahu zákazníka. K dispozici je pak záruční a

havarijní servis. Pokud se zákazník rozhodne si zavlažovací systém zapojit zcela sám, výrazně se pak snižuje celková pořizovací cena, ale zvyšuje se obtížnost a při malých zkušenostech i možnost špatného zapojení rozvodného systému či postřikovačů samotných.

Po nainstalování, jak už bylo psáno výše, zákazník nemusí vůbec do systému zasahovat a zalévání probíhá samo podle nastavení, pokud ovšem chce přenastavit řídicí jednotku, tak nepotřebuje žádné znalosti, a dokáže jednoduše řídicí jednotku přenastavit podle přiloženého manuálu. Při větší poruše je třeba provést výkopové práce a tudíž i zavolat firmu, která systém nainstalovala.

## 4 Softwarová výbava

### Ethernetové rozhraní

**Možnost dálkového přístupu byla vytvořena za těmito účely:**

- Předčít levné komerční systémy nabízené na trhu
- Zpohodlnění a zrychlení nastavování  
(nastavování přes síť zabere zlomek času co nastavování lokální stanice)
- Možnost manipulace se systémem, i na větší vzdálenost  
(například v práci nebo na dovolené; kdekoli na světě můžete provádět se systémem veškeré operace)

**Při vytváření ethernetového rozhraní bylo hlavními cíli:**

- Zajistit maximální podobnost způsobu výpisu i zápisu dat, jako je tomu přímo na lokální jednotce (LCD)
- Možnost provádět ze sítě veškeré operace, které systém dovede

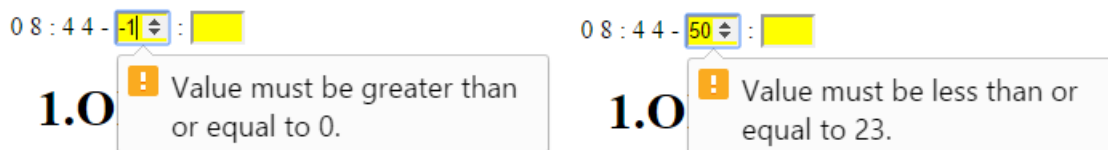
V této kapitole je popsáno celé ethernetové rozhraní s názornými ukázkami na webové stránce v prohlížeči. Řešení programových úloh naleznete v technickém koutku.

### Funkce stránky


Stránka má v hlavičce nastavené české formátování „UTF-8“ a také obnovení stránky „refresh“ (probíhá jedenkrát za minutu).

## Průběh nastavování


Nastavování je ošetřeno proti zadání nesmyslných/parazitních dat



0 8 : 4 4 -  :

1.0  Value must be greater than or equal to 0.

0 8 : 4 4 -  :

1.0  Value must be less than or equal to 23.

**Obrázek 2:** Ošetření proti zadání parazitních dat

Výběr z více možností je realizován pomocí vstupů typu ,radio‘

OUT 3 je **ZAPNUTÝ**

- ☒ ON
- ☐ OFF

```
client.print("<FORM action='http://192.168.10.145' method='GET'>");
client.print("<P> <INPUT type='radio' name='out3' value='1'>ON");
client.print("<P> <INPUT type='radio' name='out3' value='0'>OFF");
```

**Obrázek 3:** Výběr z více možností

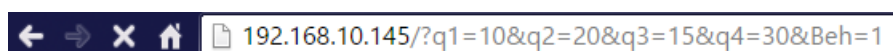
Nastavení a výběr je potřeba potvrdit tlačítkem Odeslat








**Obrázek 4:** Potvrzení nastavení

Stránka poté vrací řídicí jednotce podobný kód, který dále zpracováváme.

(Popis zpracování naleznete v technickém koutku).



     192.168.10.145/?q1=10&q2=20&q3=15&q4=30&Beh=1

**Obrázek 5:** Kód vrácený stránkou

## Vzhled jednotlivých záložek na Ethernetu doplněn o vzhled této funkce na LCD

V této kapitole budete mít možnost prohlédnout si všechny záložky, které na ethernetu můžete nalézt. Je k nim doplněn i vzhled a zpracování této funkce na LCD.

### Hlavní menu

Ethernetové rozhraní pro zavlažovací systém, Kočiča Filip ME4						
PONDĚLÍ	ÚTERÝ	STŘEDA	ČTVRTEK	PÁTEK	SOBOTA	NEDĚLE
Nastavení parametrů						Režim + Kalibrace
Vstupy						Porucha rozepnuta
Výstupy				Aktualní čas		

**Obrázek 6:** Rozložení dat v hlavním menu (Ethernet)

The image shows a blue LCD screen with white text. The text is organized into four lines. The first line shows the days of the week: Po, Ut, St, Ct, Pa, So, N. The second line shows NU=90%, NH=25%, R=1. The third line shows MU=0%, MH=100%, P=0. The fourth line shows OUT: 1, 2, 3, Ct, 22:07. The text is in a monospaced font.

◆Po	Ut	St	Ct	Pa	So	N
NU=90%		NH=25%		R=1		
MU=0%		MH=100%		P=0		
OUT: 1		2	3	Ct, 22:07		

**Obrázek 7:** Rozložení dat v hlavním menu (LCD)

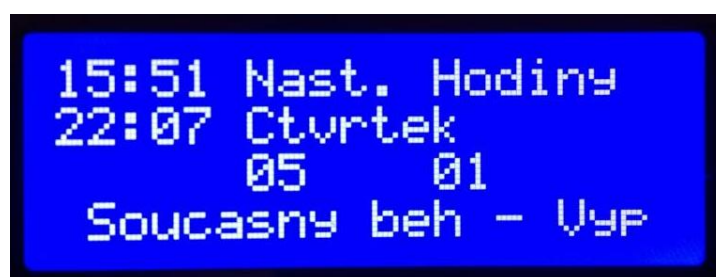


## Záložka dny (Po-Ne)

Ethernetové rozhraní pro zavlažovací systém, Kočica Filip ME4						
PONDĚLÍ	ÚTERÝ	STŘEDA	ČTVRTEK	PÁTEK	SOBOTA	NEDĚLE
Nastavení parametrů					Režim + Kalibrace	
Vstupy					Porucha rozepnuta	
Výstupy				Aktualní čas		

[Zpět](#)  
**Pondělí**  
 0 8 : 4 4 - 10 : 20  
**1.Okruh**  
 4 6 - 15  
**2.Okruh**  
 9 2 - 30  
 Současný Běh je **VYPNUTÝ**  
☒ ON ☐ OFF

**Obrázek 8:** Nastavování dní Pondělí-Neděle (Ethernet)



**Obrázek 9:** Nastavování dní Pondělí-Neděle (LCD)

## Záložka Nastavení parametrů

Zavlažovací systém Kočica Filip ME4						
PONDĚLÍ	ÚTERÝ	STŘEDA	ČTVRTEK	PÁTEK	SOBOTA	NEDĚLE
Nastavení parametrů					Režim + Kalibrace	
Vstupy					Porucha rozepnuta	
Výstupy				Aktualní čas		

[Zpět](#)

**Vlhkost (max 100)**

90 -

**Hladina (max 100)**

70 -

**Hystereze (max 10)**

1 -

**Limit Displaye**

60 -  °

**Obrázek 10:** Nastavení hlavních parametrů (Ethernet)



**Obrázek 11:** Nastavení hlavních parametrů (LCD)

## Záložka Režim + Kalibrace

Zavlažovací systém Kočica Filip ME4						
PONDĚLÍ	ÚTERÝ	STŘEDA	ČTVRTEK	PÁTEK	SOBOTA	NEDĚLE
Nastavení parametrů					Režim + Kalibrace	
Vstupy					Porucha rozepnuta	
Výstupy				Aktualní čas		

Zpět

**Nastavení Režimu + Kalibrace**

- ◉ NORMAL
- \* SUCHO
- ◉ VLHKO
  
- ◉ Kalibrace VLHKOST
- \* Kalibrace HLADINA

Submit

**Obrázek 12:** *Výběr Režimu + Kalibrace (Ethernet)*

Na LCD je tato záložka rozdělena na 2 funkce. Bývaly by zrealizovány také dvě záložky na ethernetu, ale kvůli nedostatku paměti musí být Výběr režimu i Kalibrace v jedné záložce.

<pre>* NASTAVENI REZIMU * *Rezim NORMAL      (A) Rezim SUCHO Rezim VLHKO</pre>	<pre>*KALIBRACE VLHKOSTI* * KALIBRUJ! *KALIBRACE HLADINY * KALIBRUJ!</pre>
--	--

**Obrázek 13:** *Výběr Režimu + Kalibrace (LCD)*

## Záložka Vstupy

Zavlažovací systém Kočica Filip ME4						
PONDĚLÍ	ÚTERÝ	STŘEDA	ČTVRTEK	PÁTEK	SOBOTA	NEDĚLE
Nastavení parametrů					Režim + Kalibrace	
Vstupy					Porucha rozepnuta	
Výstupy				Aktualní čas		

[Zpět](#)

Nast. Vlhkost: 90%

Měřená Vlhkost: 0%

**PODMÍNKA SPLNĚNA**

Nast. Hladina: 100%

Měřená Hladina: 100%

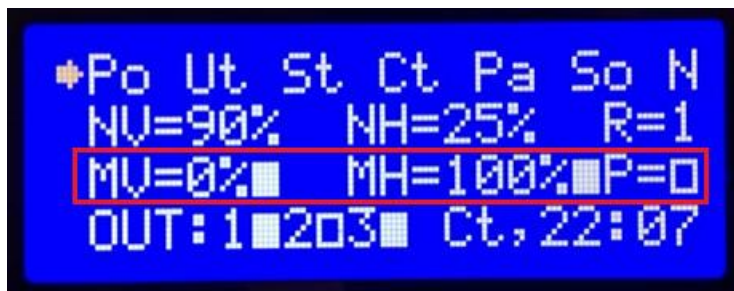
**PODMÍNKA NESPLNĚNA**

Hystereze: 0%

Limit Displaye: 60m

**Obrázek 14:** Vstupy (Ethernet)

Na LCD jsou vstupy zobrazeny přímo na hlavní obrazovce, není potřeba vstupovat do žádné funkce. (Měřená Vlhkost = 0 %; Měřená Hladina = 100 %; Porucha – neaktivní)

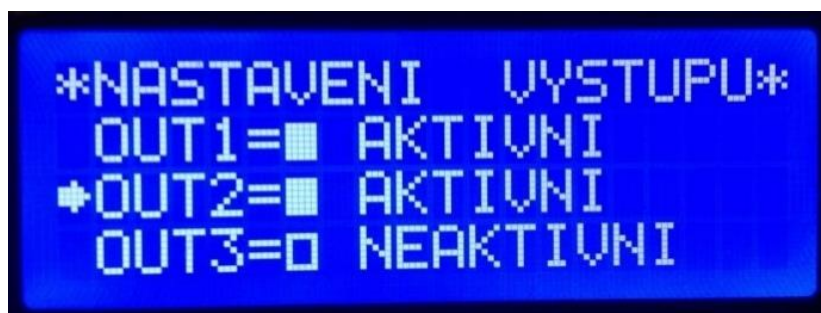


**Obrázek 15:** Vstupy (LCD)

## Záložka Výstupy

Vstupy		Porucha rozepnuta
Výstupy	Aktualní čas	
<input type="button" value="Zpět"/>		
OUT 1 je <b>ZAPNUTÝ</b>		
OUT 2 je <b>VYPNUTÝ</b>		
<input type="radio"/> ON		
<input type="radio"/> OFF		
<input type="button" value="Submit"/>		
OUT 3 je <b>ZAPNUTÝ</b>		
<input type="radio"/> ON		
<input type="radio"/> OFF		
<input type="button" value="Submit"/>		

**Obrázek 16:** Manuální spouštění výstupů (Ethernet)



**Obrázek 17:** Manuální spouštění výstupů (LCD)

## Záložka Aktuální čas

Zavlažovací systém Kočica Filip ME4						
PONDĚLÍ	ÚTERÝ	STŘEDA	ČTVRTEK	PÁTEK	SOBOTA	NEDĚLE
Nastavení parametrů					Režim + Kalibrace	
Vstupy					Porucha rozepnuta	
Výstupy				Aktuální čas		

[Zpět](#)

Středa, 14 : 8 : 52 .

12 : 25

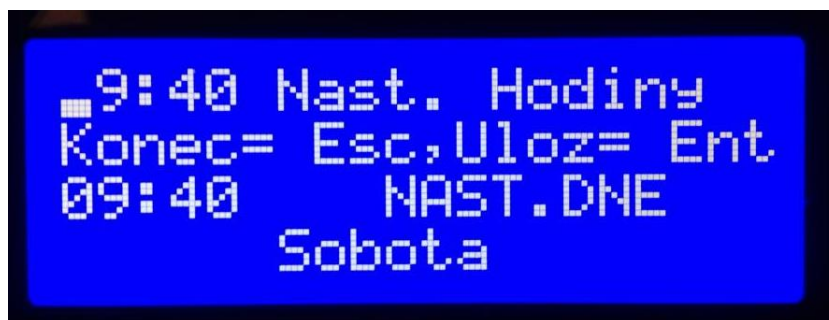
• Po

• Út

• St

• Čt

**Obrázek 18:** Aktuální čas (Ethernet)



**Obrázek 19:** Aktuální čas (LCD)

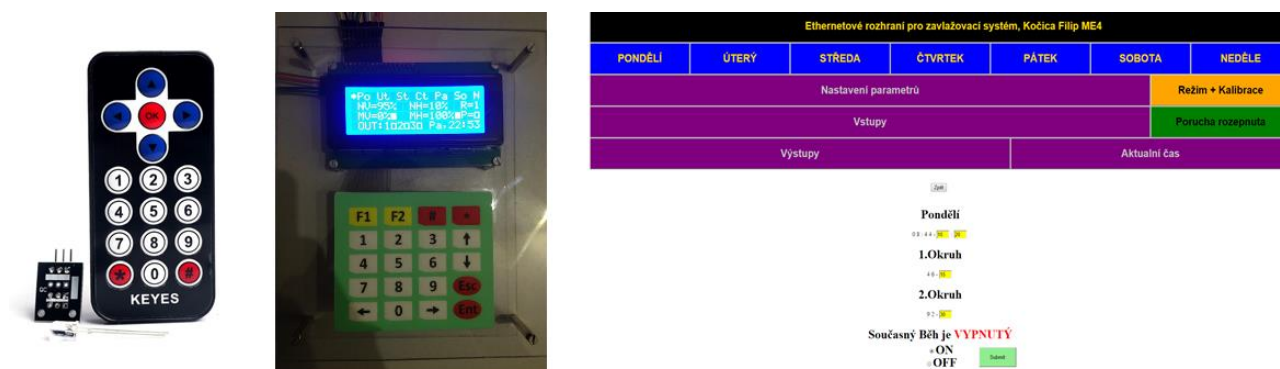
## Řídicí jednotka a program

V této kapitole nejsou uváděny úryvky ze zdrojového kódu řídicího programu, ale pro technické nadšence jsem ty nejdůležitější úryvky doplnil o popis a umístil je do technického koutku. Tyto části kódu byly upraveny (zkráceny), a to pro lepší názornost ukázek.

Program byl vytvořen 16. května 2015 ve vývojovém prostředí Arduino 1.6.6, celkem čítá přes 4200 řádků, zabírá 71 kB (28 % paměti pro program) a 7,5 kB (91 % statické paměti pro data).

## Možnosti řízení systému

- Lokálně, a to buď pomocí LCD 20 × 4 s membránovou klávesnicí 5 × 4 nebo infračerveným dálkovým ovládáním
- Vzdáleně, přes ethernetové rozhraní

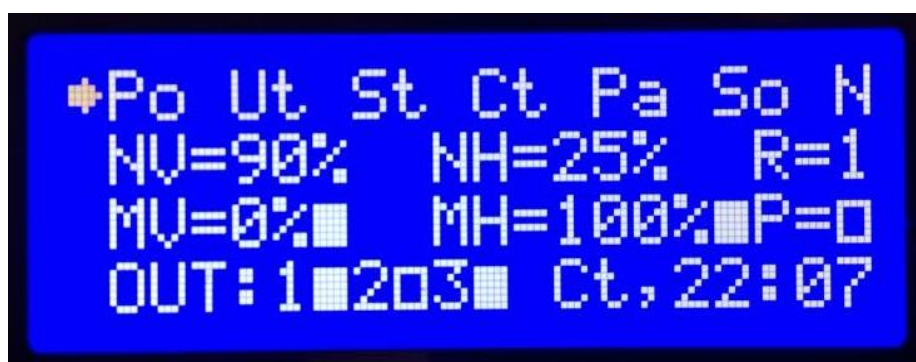


Obrázek 20: Možnosti řízení systému

## Rozložení dat na displeji

Při rozmisťování dat na display byl ve všech funkcích kladen důraz na co nejvyšší čitelnost a srozumitelnost.

- Pohybovat v menu se lze kurzorem ve tvaru šipky a skok do funkce je uskutečněn přes tlačítko ENTER.
- Na prvním řádku menu jsou dny Po-Ne
- Na druhém se zobrazují námi nastavené veličiny a aktuální režim
- Na třetím se zobrazují hodnoty z elektrod, ultrazvukového a dešťového senzoru.
- A na čtvrtém jsou výstupy a aktuální den a čas
- Obdélníky u měřených parametrů signalizují, zda je podmínka splněna (plný – splněna; prázdný – nesplněna); u poruchy, zda je aktivní (plný – aktivní; prázdný – neaktivní); u výstupů, zda jsou v provozu (plný – je v provozu; prázdný – není v provozu)



**Obrázek 21:** Rozložení dat v hlavním menu

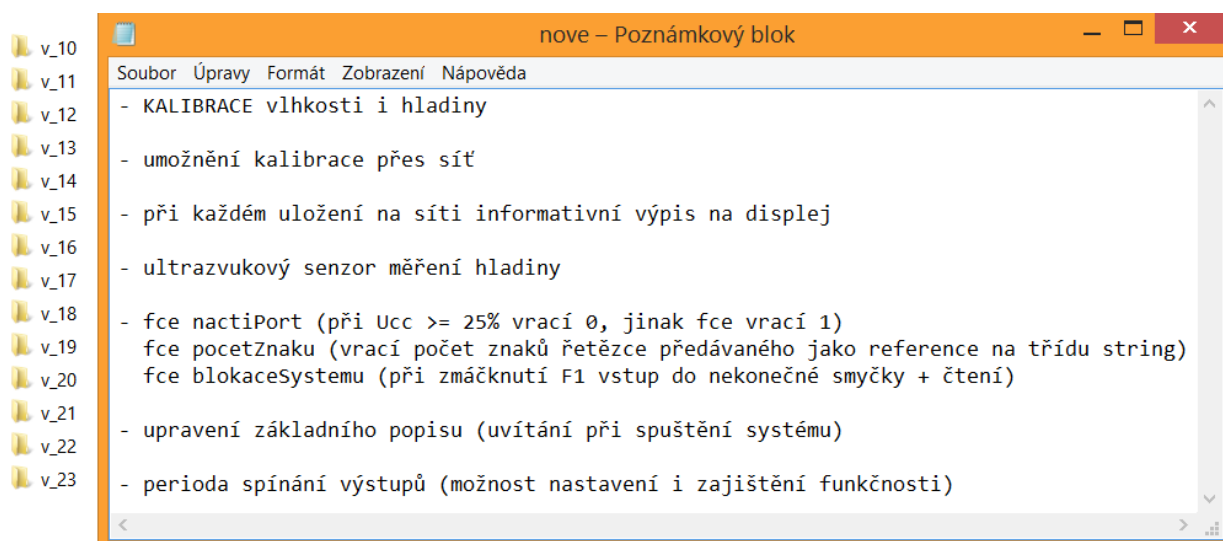


## Základní struktura programu

Struktura programu je vytvořena hlavně v závislosti na tom, že řídicí jednotka má tři na sobě nezávislá ovládací rozhraní a že je potřeba provádět naplánované procesy v reálném čase (spouštět výstupy v přesně nastavený čas, číst všechny vstupy, číst z klávesnice, z ovladače i ze sítě, vypisovat data na LCD i na síť) nezávisle na stavu ovládacích rozhraní.

Proto je program rozdělen na hlavní funkce, které se starají o jednotlivé procesy.

Řídicí program jsem vytvářel dlouho a nejdennokrát jsem se musel podívat, nebo dokonce se vrátit, na předchozí verze programů (například když si systém začal dělat, co chtěl, a nemohl jsem najít jádro problému), proto si při každé velké úpravě vytvořím další verzi a připsu si do textového souboru, co jsem kde přidal/upravil/smazal. Tento systém verzí mi hodně pomohl při vytváření i při obhajobě.

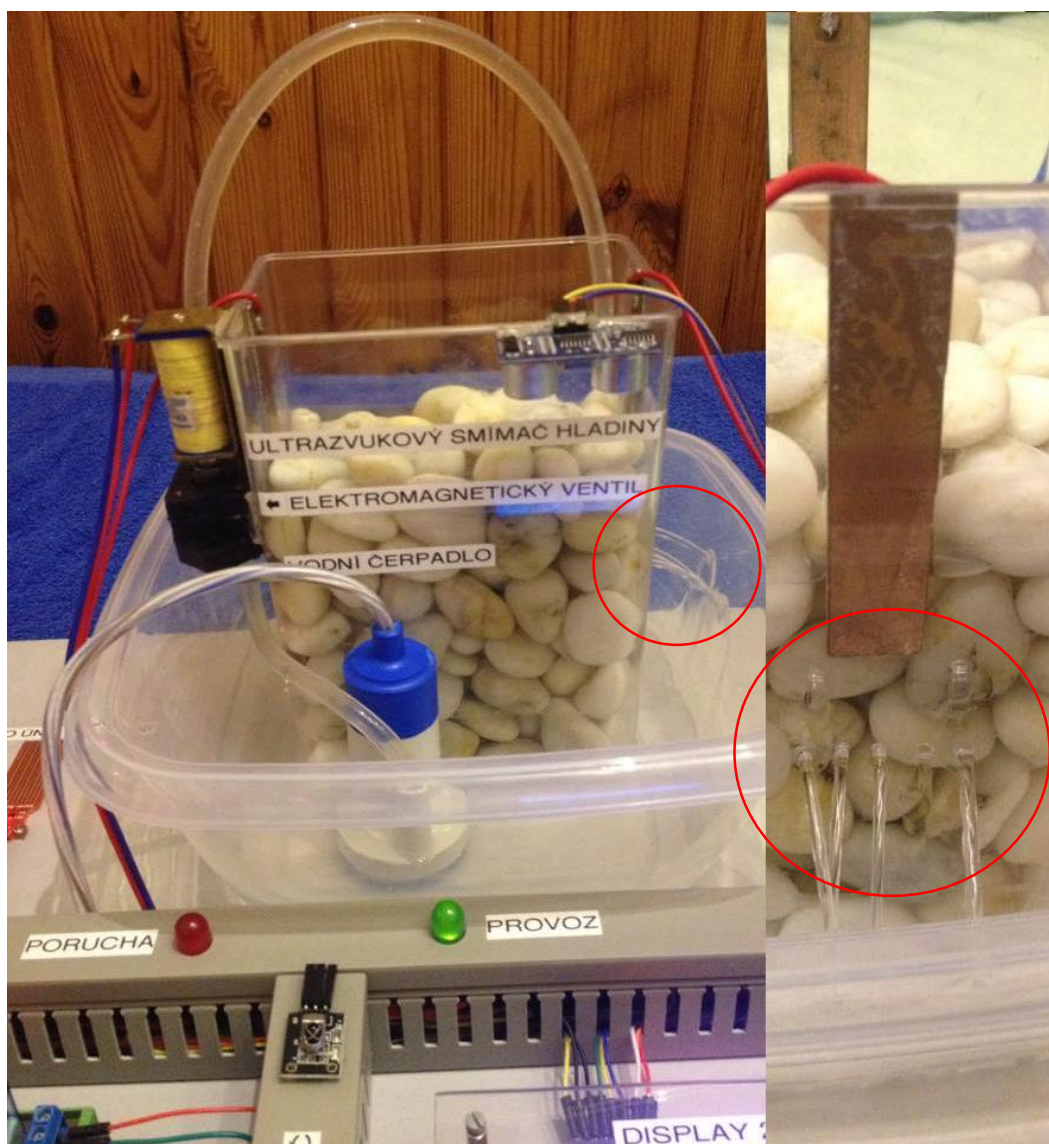


**Obrázek 22:** Verze programu + nově přidané do programu

Více o programu se dozvíte v **technickém koutku**, kde jsem pro Vás podrobně popsal nejdůležitější procesy.

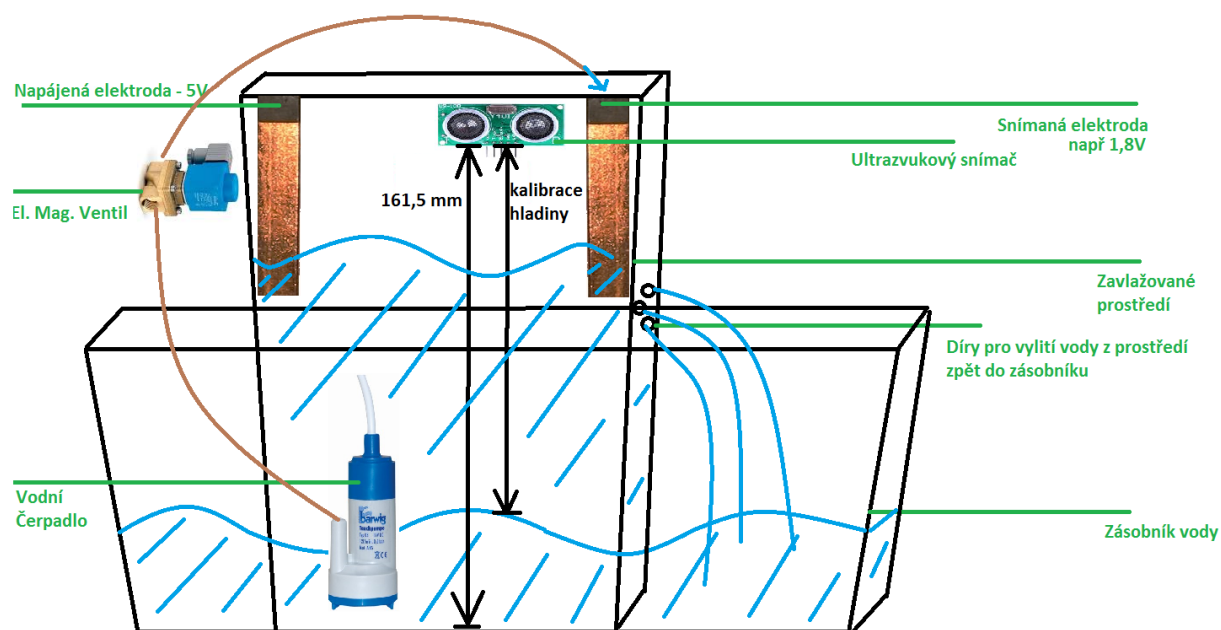
## 5 Fyzická realizace

Realizace simulace zavlažování prošla dlouhým a složitým vývojem. Ze začátku jsem měl dvě nádoby a jedno čerpadlo, kterým jsem proměňoval vodu v obou nádobách, ale vzhledem k neefektivnosti jsem od tohoto řešení upustil a vymyslel jsem automatický systém, kdy se voda napuštěná do zavlažovaného prostředí po čase sama vyleje vyvrtanými dírami zpět do zásobníku vody (simulující vypařování vody z prostředí a fyzické doplnění vody do nádrže).



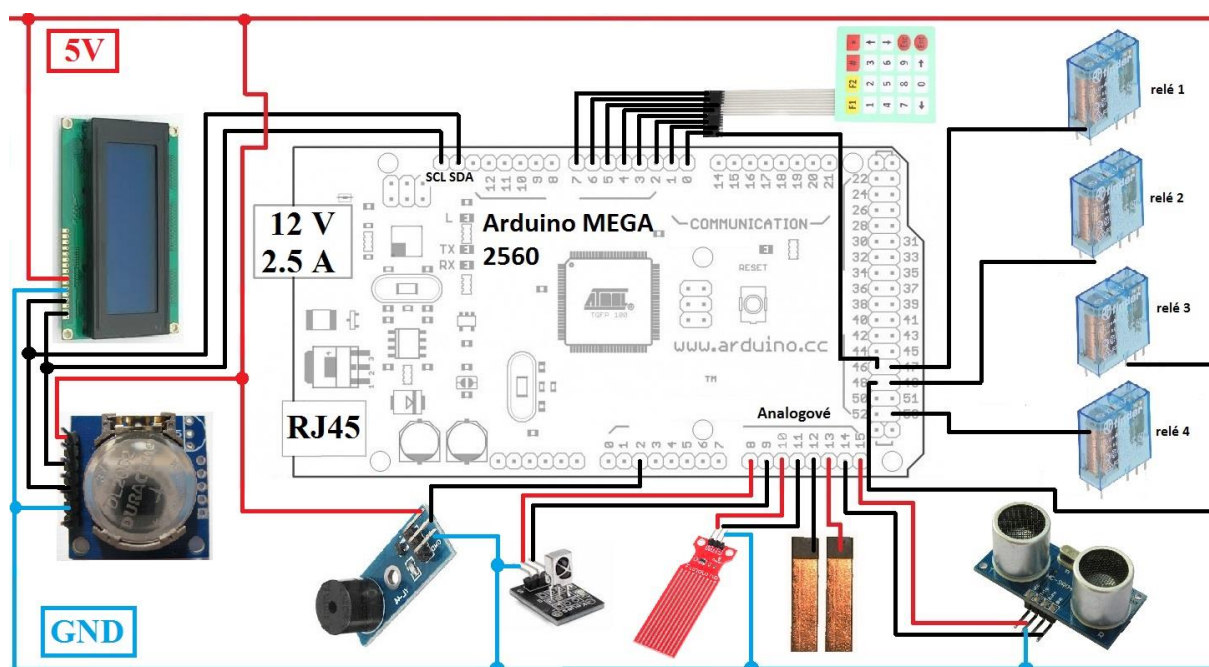
**Obrázek 23:** Fyzická realizace zavlažování

## Popis vytvořené simulace zavlažování



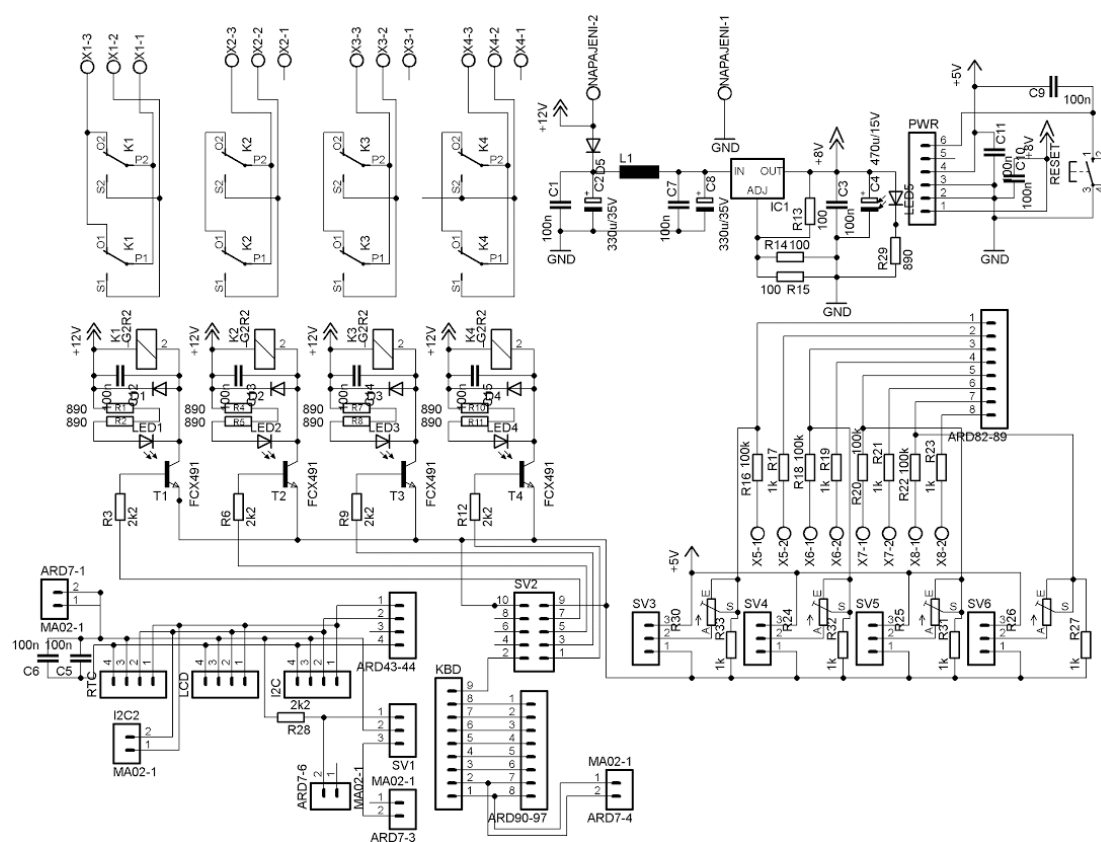
Obrázek 24: Popis vytvořené simulace zavlažování

## Blokové schéma

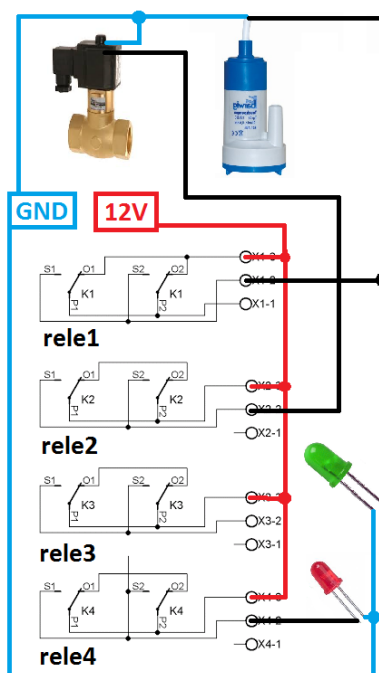


Obrázek 25: Blokové schéma

## Schéma zapojení

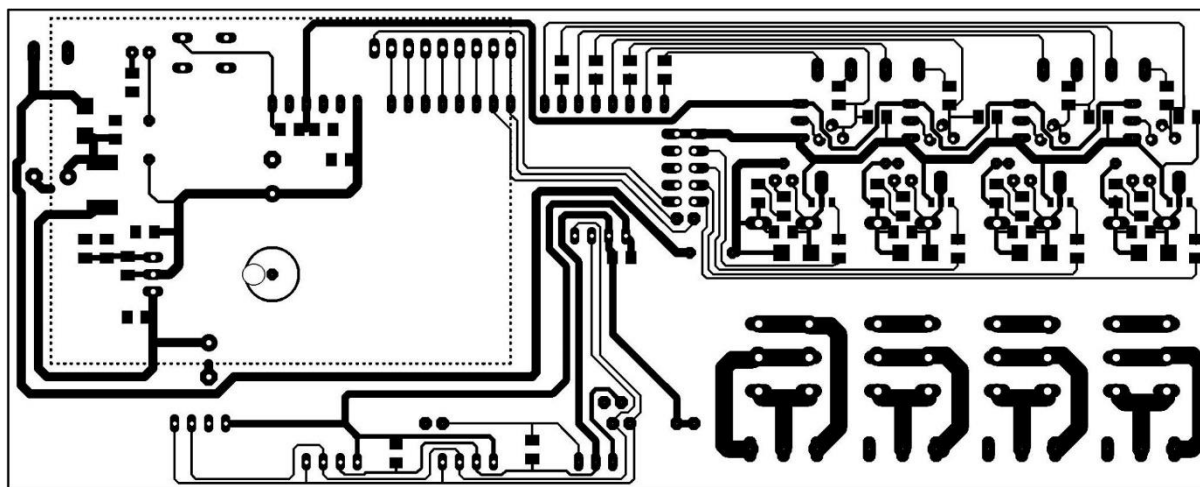


Obrázek 26: Schéma zapojení



Obrázek 27: Zapojení elektromagnetických relé

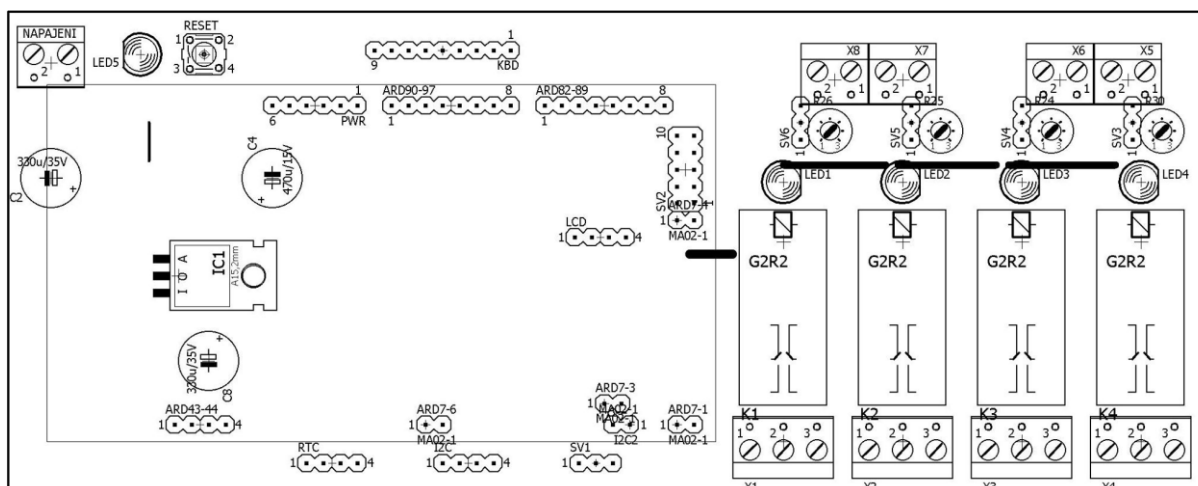
## Plošný spoj



Obrázek 28: Plošný spoj

## Osazovací plán

Pohled ze strany součástek



Obrázek 29: Osazovací plán – pohled ze strany součástek

## 6 Technický koutek

V této části se mohou techničtí nadšenci dozvědět mnoho o řídicím programu systému. To, jak jsou řešeny nejdůležitější procesy a počáteční výběr datových struktur.

### Třída Zapouzdření dat

Na začátku programu deklarujeme třídu, která v privátní části definuje hlavní proměnné a ve veřejné části metody, které budeme dále v programu využívat. Je vytvořeno pole struktur „dny“, struktury „Nastavení Dny“ o sedmi prvcích právě proto, aby každý ze sedmi dní měl své vlastní nastavení.

```
class Zapouzdreni_dat
{
private:
    int limitdispaye = NULL, nastvlhkosti = NULL, nasthladiny = NULL, nasthystereze = NULL, kalibraceVlhkost = NULL, kalibraceHladina = NULL;

    typedef struct {                //struktura pole
        int hodiny1;
        int hodiny2;
        int hodinycelk;
        int minuty1;                // PRIVATE: (ZAPOUZDŘENÉ PROMĚNNÉ)
        int minuty2;                // LZE S NIMI OPEROVAT POUZE POMOCÍ ČLENSKÝCH FCÍ / METOD
        int minutycelk;
        int min1;
        int min10;
        int min1celk;
        int min2;
        int min20;
        int min2celk;
    }
    nastaveniDny;
    nastaveniDny dny[6];            //strukturové pole

    int nastavORC[10] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

public:                            // METODY / ČLENSKÉ FCE TŘÍDY
    Zapouzdreni_dat();              //KONSTRUKTOR
    int nastaveniParametru(int limitdispaye, int nasthladiny, int nastvlhkosti, int nasthystereze); //fce pro nas
    int cas(int limitdispaye);      //fce pro nas
    int nastavcas(int limitdispaye); //fce pro nas
    int nastavvystupy(int limitdispaye); //fce pro nas
    int vyberRezim();              //fce pro vyt
    int hlaseniPoruchy(bool nastavvystupy1, bool nastavvystupy2, int Binarnivstup1, int i); //fce pro hlá

    int zadaniHesla(int adresaPameti, int zadejNeboZmen); //slouží pro
    void nactiVstupy(int analogvstup1, int analogvstup2, int cteniVstupu1, int cteniVstupu2); //procedura r
    void kalibrace();              //nakalibruje
    void limit();                 //procedura t
    void prectiEEPROM();          //procedura,
    void zapisEEPROM();           //procedura,
    void prace_s_vystupy();       //procedura,
    void zobrazNaDisplayi();      //procedura,
    void stlaceniTlacitkaLoop(); //procedura,
    void ethernet();              //tato proced
    ~Zapouzdreni_dat();          //DESTRUKTOR
};
```

Obrázek 30: Třída Zapouzdření dat



## Funkce loop

Funkce loop na začátku provede inicializaci objektu třídy pomocí operátoru new, zkontrolování paměti EEPROM, načtení vstupů z pinů uvedených v signatuře, výpis základních informací o systému na displej a pak program vstoupí do nekonečné smyčky, kde se pořád dokola volají funkce, které řídí celý systém.

```
void loop()
{
    Zapouzdeni_dat * den = new Zapouzdeni_dat; // VYTVOŘENÍ D
    privitani(5000); // vypíše info
    //den->zadaniHesla(9, 1); // zavolá pr
    den->nactiVstup(A14, A12, A15, A13); // funkce kter
    zkontrolujEEPROM(); // procedura z
    for (;;) { // nekonečná s
        den->prectiEEPROM(); // načte dat
        den->prace_s_vystupy(); // kontroluj
        den->zobrazNaDisplayi(); // Zobrazí n
        problikavani(); // zajišťuje
        den->limit(); // vypnutí p
        Binarnivstup1 = nactiPort(A11); // fce vrátí
        if ((!Binarnivstup1) && (!pozastavPoruchu))
            den->hlaseniPoruchy(nastavvystupy1, nastavvystupy2, Binarnivstup1, i); // zavolá
        den->stlaceniTlacitkaLoop(); // Snímá stl
        den->ethernet(); // zajišťuje
    }
    delete[] den; // uvolnění pa
}
```

Obrázek 31: Funkce loop

## Procedura zápis do EEPROM

Procedura zápis do EEPROM se volá při každém nastavení a nastavené proměnné zapíše do paměti, každá z proměnných má své místo v paměti o kapacitě 4 kB, která je pro tyto účely plně dostačující. Zadávání je rovněž ošetřeno proti zadání nesmyslných parazitních údajů.

```
void Zapouzdeni_dat::zapisEEPROM()
{
    if ((nasthladiny >= NULL) && (nasthladiny <= 100)) EEPROM.write(0 + rezimEEPROM, nasthladiny);
    if ((nastvlhkosti >= NULL) && (nastvlhkosti <= 100)) EEPROM.write(1 + rezimEEPROM, nastvlhkosti);
    if ((limitdispaye >= NULL) && (limitdispaye <= 60)) EEPROM.write(2 + rezimEEPROM, limitdispaye);
    if ((nedele2 >= NULL) && (nedele2 <= 9)) EEPROM.write(3 + rezimEEPROM, nedele2);
    if ((soucasnyBeh >= NULL) && (soucasnyBeh <= 1)) EEPROM.write(4 + rezimEEPROM, soucasnyBeh);
    if ((nasthystereze >= NULL) && (nasthystereze <= 10)) EEPROM.write(5 + rezimEEPROM, nasthystereze);
    if ((kalibraceVlhkost >= NULL) && (kalibraceVlhkost <= 1023)) EEPROM.write(6 + rezimEEPROM, kalibraceVlhkost);
    if ((kalibraceHladina >= NULL) && (kalibraceHladina <= 1023)) EEPROM.write(7 + rezimEEPROM, kalibraceHladina);
    if ((periodaSpousteni >= NULL) && (periodaSpousteni <= 99)) EEPROM.write(8 + rezimEEPROM, periodaSpousteni);

    int c = 11 + rezimEEPROM;
    for (int a = NULL; a < 7; a++) {
        if ((dny[a].hodiny1 >= NULL) && (dny[a].hodiny1 <= 2)) EEPROM.write(c, dny[a].hodiny1);
        c++;
        if ((dny[a].hodiny2 >= NULL) && (dny[a].hodiny2 <= 9)) EEPROM.write(c, dny[a].hodiny2);
        c++;
        if ((dny[a].hodinycelk >= NULL) && (dny[a].hodinycelk <= 23)) EEPROM.write(c, dny[a].hodinycelk);
        c++;
        if ((dny[a].minuty1 >= NULL) && (dny[a].minuty1 <= 5)) EEPROM.write(c, dny[a].minuty1);
        c++;
        if ((dny[a].minuty2 >= NULL) && (dny[a].minuty2 <= 9)) EEPROM.write(c, dny[a].minuty2);
        c++;
        if ((dny[a].minutycelk >= NULL) && (dny[a].minutycelk <= 59)) EEPROM.write(c, dny[a].minutycelk);
        c++;
        if ((dny[a].min1 >= NULL) && (dny[a].min1 <= 9)) EEPROM.write(c, dny[a].min1);
        c++;
        if ((dny[a].min10 >= NULL) && (dny[a].min10 <= 9)) EEPROM.write(c, dny[a].min10);
        c++;
        if ((dny[a].min1celk >= NULL) && (dny[a].min1celk <= 99)) EEPROM.write(c, dny[a].min1celk);
        c++;
        if ((dny[a].min2 >= NULL) && (dny[a].min2 <= 9)) EEPROM.write(c, dny[a].min2);
        c++;
        if (a != 6) {
            if ((dny[a].min20 >= NULL) && (dny[a].min20 <= 9)) EEPROM.write(c, dny[a].min20);
        }
        c++;
        if ((dny[a].min2celk >= NULL) && (dny[a].min2celk <= 99)) EEPROM.write(c, dny[a].min2celk);
        c++;
    }
}
```

Obrázek 32: Procedura zápis do EEPROM



## Princip zpracování dat načtených z ethernetu do vyrovnávací paměti

Řekněme, že máme na ethernetu datovou strukturu „number“-je to rámeček, do kterého se zadává číselná hodnota a bude se jmenovat „q1“. Cyklus „for“ prohledává řetězec z ethernetu od počátku po poslední znak, který spočítá funkce počet Znaků, přebírající referenci na tento řetězec třídy string, až narazí na řetězec ve tvaru: „q1=číslo&“, vloží to, co je na pozici toho čísla (číslo samotné), a umístí ho do příslušné proměnné, samozřejmě s přepočtem v ASCII tabulce, poté zavolá proceduru zápis do EEPROM a vypíše na LCD, že probíhá zápis do paměti EEPROM. Pro dvojciferné číslo je tvar následující: „q1=čísločíslo&“.





















```
for (int v = NULL; v <= pocetZnaku(buffer); v++) {  
  
    //-----  
    if ((buffer[v] == 'q') && (buffer[v + 1] == '1') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] == '&')) { // hodiny 2  
        dny[vypisDniNaSit].hodiny1 = NULL;  
        dny[vypisDniNaSit].hodiny2 = ((int)buffer[v + 3] - 48);  
        dny[vypisDniNaSit].hodinycelk = ((int)buffer[v + 3] - 48);  
        zapisEEPROM();  
        if (vypisMozny) nastaveniNaEthernetu(2000, 0);  
    }  
  
    if ((buffer[v] == 'q') && (buffer[v + 1] == '1') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] != '&') && (buffer[v + 5] == '&')) {  
        dny[vypisDniNaSit].hodiny1 = ((int)buffer[v + 3] - 48);  
        dny[vypisDniNaSit].hodiny2 = ((int)buffer[v + 4] - 48);  
        dny[vypisDniNaSit].hodinycelk = ((10 * ((int)buffer[v + 3] - 48)) + ((int)buffer[v + 4] - 48));  
        zapisEEPROM();  
        if (vypisMozny) nastaveniNaEthernetu(2000, 0);  
    }  
  
    //-----  
    if ((buffer[v] == 'q') && (buffer[v + 1] == '2') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] == '&')) { // minuty 2  
        dny[vypisDniNaSit].minuty1 = NULL;  
        dny[vypisDniNaSit].minuty2 = ((int)buffer[v + 3] - 48);  
        dny[vypisDniNaSit].minutycelk = ((int)buffer[v + 3] - 48);  
        zapisEEPROM();  
        if (vypisMozny) nastaveniNaEthernetu(2000, 0);  
    }  
  
    if ((buffer[v] == 'q') && (buffer[v + 1] == '2') && (buffer[v + 2] == '=') && (buffer[v + 3] != '&') && (buffer[v + 4] != '&') && (buffer[v + 5] == '&')) {  
        dny[vypisDniNaSit].minuty1 = ((int)buffer[v + 3] - 48);  
        dny[vypisDniNaSit].minuty2 = ((int)buffer[v + 4] - 48);  
        dny[vypisDniNaSit].minutycelk = ((10 * ((int)buffer[v + 3] - 48)) + ((int)buffer[v + 4] - 48));  
        zapisEEPROM();  
        if (vypisMozny) nastaveniNaEthernetu(2000, 0);  
    }  
}  
  
int pocetZnaku(string & retezec) {  
    int pocet = NULL;  
    for (; retezec.c_str()[pocet] != '\0';)  
        pocet++;  
    return pocet;  
}
```

**Obrázek 33:** Princip zpracování dat načtených z ethernetu do vyrovnávací paměti

## 7 Stručný výčet všech funkcí systému

- Zadání a změna hesla, při změně parametrů je potřeba zadat heslo
- Manuální blokace systému
- Snímání a hlášení poruchy systému, možnost pozastavení poruchy
- Měření výšky hladiny vody v nádrži a vlhkosti zavlažovaného prostředí
- Nastavení vlhkosti, hladiny, hystereze, limitu displaye, obvodu reálného času (Den, Hodiny, Minuty), dní Pondělí - Neděle (Hodiny, Minuty, doba trvání sepnutí 1. a 2. okruhu, současný běh a periodu spouštění výstupů)
- Manuální spouštění výstupů
- Výběr režimu (Normal; Sucho; Vlhko)
- Kalibrace hladiny i vlhkosti
- Komunikace přes síť
- Možnost ovládání pomocí dálkového ovladače a infračerveného přijímače

## 8 Porovnání této práce s reálným komerčním systémem nabízeným na trhu

	Běžný komerční systém	Tato práce
Možnost vzdáleného přístupu		
Indikace probíhající závlahy		
Nastavení délky závlahy		
Více startů denně		
Sedmidenní zavlažovací kalendář s volbou zavlažovacích dní		
Volba zavlažování sudé/liché dny		
Manuální spouštění sekcí		
Manuální blokování programu		
Manuální zadání přestávky v závlaze 1–31 dní		
Zabezpečení		

## Závěr

Vytvořený model splňuje všechny požadavky stanovené na začátku práce. Je plně funkční a zcela vystihuje veškeré možné situace v zahradách. Důraz byl kladený především na názornost celého modelu a zároveň na využití automatizace jako nedílné součásti všech moderních systémů využívaných v dnešní době.

Práce pro mě byla výzvou, protože automatizace systémů, které pracují ve venkovním prostředí, je jednou z nejsložitějších, neboť zde existuje spousta proměnných, které se nedají předpovídat, například počasí, vlhkost, teplota atd.

S výsledkem své práce jsem spokojen a touto tematikou se stále zabývám.

## Použitá literatura a zdroje

1. PRATA, Stephen. *Mistrovství v C++*. 4. aktualizované vyd. Brno: COMPUTER PRESS, 2013. ISBN 978-80-251-3828-1.
2. Zavlažovací Systémy. In: *azzahrada.cz – Zavlažovací systémy a závlahy* [online]. 2010. vyd. [cit. 2010-12-14]. Dostupné z: <azzahrada.cz>
3. ENDRYCH, Václav. *Elektronický systém zavlažování*. Brno, 2011. VUT.

# Seznam obrázků

<b>OBRÁZEK 1:</b> FOTOGRAFIE VÝSLEDNÉHO VÝROBKU .....	9
<b>OBRÁZEK 2:</b> OŠETŘENÍ PROTI ZADÁNÍ PARAZITNÍCH DAT .....	14
<b>OBRÁZEK 3:</b> VÝBĚR Z VÍCE MOŽNOSTÍ .....	14
<b>OBRÁZEK 4:</b> POTVRZENÍ NASTAVENÍ .....	14
<b>OBRÁZEK 5:</b> KÓD VRÁCENÝ STRÁNKOU .....	14
<b>OBRÁZEK 6:</b> ROZLOŽENÍ DAT V HLAVNÍM MENU (ETHERNET) .....	15
<b>OBRÁZEK 7:</b> ROZLOŽENÍ DAT V HLAVNÍM MENU (LCD) .....	15
<b>OBRÁZEK 8:</b> NASTAVOVÁNÍ DNÍ PONDĚLÍ-NEDĚLE (ETHERNET) .....	16
<b>OBRÁZEK 9:</b> NASTAVOVÁNÍ DNÍ PONDĚLÍ-NEDĚLE (LCD) .....	16
<b>OBRÁZEK 10:</b> NASTAVENÍ HLAVNÍCH PARAMETRŮ (ETHERNET) .....	17
<b>OBRÁZEK 11:</b> NASTAVENÍ HLAVNÍCH PARAMETRŮ (LCD) .....	17
<b>OBRÁZEK 12:</b> VÝBĚR REŽIMU + KALIBRACE (ETHERNET) .....	18
<b>OBRÁZEK 13:</b> VÝBĚR REŽIMU + KALIBRACE (LCD) .....	18
<b>OBRÁZEK 14:</b> VSTUPY (ETHERNET) .....	19
<b>OBRÁZEK 15:</b> VSTUPY (LCD) .....	19
<b>OBRÁZEK 16:</b> MANUÁLNÍ SPOUŠTĚNÍ VÝSTUPŮ (ETHERNET) .....	20
<b>OBRÁZEK 17:</b> MANUÁLNÍ SPOUŠTĚNÍ VÝSTUPŮ (LCD) .....	20
<b>OBRÁZEK 18:</b> AKTUÁLNÍ ČAS (ETHERNET) .....	21
<b>OBRÁZEK 19:</b> AKTUÁLNÍ ČAS (LCD) .....	21
<b>OBRÁZEK 20:</b> MOŽNOSTI ŘÍZENÍ SYSTÉMU .....	22
<b>OBRÁZEK 21:</b> ROZLOŽENÍ DAT V HLAVNÍM MENU .....	23
<b>OBRÁZEK 22:</b> VERZE PROGRAMU + NOVĚ PŘIDANÉ DO PROGRAMU .....	24
<b>OBRÁZEK 23:</b> FYZICKÁ REALIZACE ZAVLAŽOVÁNÍ .....	25
<b>OBRÁZEK 24:</b> POPIS VYTVOŘENÉ SIMULACE ZAVLAŽOVÁNÍ .....	26
<b>OBRÁZEK 25:</b> BLOKOVÉ SCHÉMA .....	26
<b>OBRÁZEK 26:</b> SCHÉMA ZAPOJENÍ .....	27
<b>OBRÁZEK 27:</b> ZAPOJENÍ ELEKTROMAGNETICKÝCH RELÉ .....	27
<b>OBRÁZEK 28:</b> PLOŠNÝ SPOJ .....	28
<b>OBRÁZEK 29:</b> OSAZOVACÍ PLÁN – POHLED ZE STRANY SOUČÁSTEK .....	28
<b>OBRÁZEK 30:</b> TŘÍDA ZAPOUZDŘENÍ DAT .....	29
<b>OBRÁZEK 31:</b> FUNKCE LOOP .....	30
<b>OBRÁZEK 32:</b> PROCEDURA ZÁPIS DO EEPROM .....	31
<b>OBRÁZEK 33:</b> PRINCIP ZPRACOVÁNÍ DAT NAČTENÝCH Z ETHERNETU DO VYROVNÁVACÍ PAMĚTI .....	32