

Automatický zavlažovací systém

David Bábek

Bakalářská práce
2021



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **David Bábek**
Osobní číslo: **A17208**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Inteligentní systémy s roboty**
Forma studia: **Kombinovaná**
Téma práce: **Automatický zavlažovací systém**
Téma práce anglicky: **Automatic irrigation system**

Zásady pro vypracování

1. Zpracujte literární rešerši týkající se snímačů vlhkosti, akčních členů a řídicích systémů.
2. Analyzujte možnosti automatizace zavlažovacího systému.
3. Navrhněte rozhraní, které umožní připojení k řídicímu systému.
4. Napište program pro diagnostiku připojení a správné funkce jednotlivých prvků.
5. Navrhněte algoritmus a program pro automatický zavlažovací systém.
6. Zhodnotte dosažené výsledky a navrhněte možné rozšíření tohoto modelu.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. SCHMID, Dietmar. *Řízení a regulace pro strojírenství a mechatroniku*. Přeložil Jiří HANDLÍŘ. Praha: Europa-Sobotáles cz., 2005. ISBN 80-86706-10-9.
2. HÄBERLE, Heinz O. *Průmyslová elektronika a informační technologie*. Přeložil Jiří HANDLÍŘ. Praha: Europa-Sobotáles cz., 2003. ISBN 80-86706-04-4.
3. ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. *PLC a automatizace*. 1. díl, Základní pojmy, úvod do programování. Praha: BEN – technická literatura, 1999. ISBN 80-86056-58-9.
4. MAIXNER, Ladislav. *Mechatronika: učebnice*. Brno: Computer Press, 2006. ISBN 80-251-1299-3.
5. Automa – časopis pro automatizační techniku
6. KOBOLD Messring GmbH – firemní dokumentace

Vedoucí bakalářské práce: **prof. Ing. Vladimír Vašek, CSc.**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **15. ledna 2021**

Termín odevzdání bakalářské práce: **17. května 2021**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan



prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne 15. ledna 2021

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl jsem seznámen s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor;
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

.....
podpis diplomanta

ABSTRAKT

Cílem práce je navrhnout a sestavit vlastní algoritmus pro řízení a pokud možno využít dostupné snímače a akční členy s nízkou pořizovací cenou. Dále se práce věnuje dostupným řešením závlah a jejich komerčních řešením na trhu. Výsledkem práce je funkční systém, který disponuje uživatelským prostředím v podobě webové stránky, díky které lze systém konfigurovat a nastavit požadované časy závlahy. Systém zavlažování běží na platformě ESP32 (arduino), které se může programovat v prostředí Arduino IDE, které je rozepsáno v praktické části. Praktická část obsahuje i dostupné režimy funkcionality systému a možné doplnění systému o další snímače.

Klíčová slova: zavlažování, arduino, ESP, automatika, snímače

ABSTRACT

The aim of this work is to design and build your own algorithm for control and, if possible, use available sensors and actuators with a low purchase price. Furthermore, the work deals with available irrigation solutions and their commercial solutions on the market. The result of the work is a functional system that has a user interface in the form of a web page, thanks to which the system can be configured and the required irrigation times can be set. The irrigation system runs on the ESP32 (arduino) platform, which can be programmed in the Arduino IDE, which is described in the practical part. The practical part also contains the available modes of system functionality and possible addition of other sensors to the system.

Keywords: irrigation, arduino, ESP, automatic, sensors

Chtěl bych poděkovat Ing. Josefu Kováři za odborné konzultace a rady při zpracovávání této práce. Dále bych chtěl poděkovat svojí rodině za podporu během mého studia.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

| | |
|--|-----------|
| ÚVOD | 9 |
| I TEORETICKÁ ČÁST | 10 |
| 1 POPIS SNÍMAČŮ, AKČNÍCH ČLENŮ A ŘÍDÍCÍCH SYSTÉMŮ | 11 |
| 1.1 SNÍMAČE | 11 |
| 1.1.1 Snímač teploty..... | 11 |
| 1.1.2 Snímač deště..... | 12 |
| 1.1.3 Snímač půdní vlhkosti..... | 13 |
| 1.1.4 Snímač průtoku kapaliny..... | 14 |
| 1.2 AKČNÍ ČLENY | 15 |
| 1.2.1 Elektromagnetické relé..... | 15 |
| 1.2.2 Elektromagnetické ventily..... | 15 |
| 1.2.3 Čerpadlo | 16 |
| 1.3 ZOBRAZOVACÍ ČLENY | 17 |
| 1.3.1 Zobrazení na LCD..... | 17 |
| 1.3.2 Zobrazení OLED..... | 17 |
| 1.3.3 Zobrazení pomocí webového serveru | 18 |
| 1.3.4 Zobrazení pomocí LED diod..... | 18 |
| 1.4 ŘÍDÍCÍ SYSTÉM | 18 |
| 1.4.1 Arduino UNO..... | 19 |
| 1.4.2 WeMos D1 R32..... | 19 |
| 2 MOŽNOSTI AUTOMATIZACE ZAVLAŽOVACÍHO SYSTÉMU | 21 |
| 2.1 METODY ZÁVLAHY..... | 21 |
| 2.1.1 Metoda zavodňování | 21 |
| 2.1.2 Metoda postřiku..... | 21 |
| 2.1.3 Metoda kapkové závlahy..... | 22 |
| 2.2 MOŽNOSTI NA TRHU | 22 |
| 2.2.1 Gardena | 22 |
| 2.2.1.1 Zavlažovací počítače..... | 23 |
| 2.2.1.2 Příslušenství Gardena | 25 |
| 3 NÁVRH ROZHRANÍ ŘÍDÍCÍHO SYSTÉMU | 26 |
| 3.1 ŘÍDÍCÍ ČÁST..... | 26 |
| 3.2 SENZOROVÁ ČÁST | 26 |
| 3.3 AKČNÍ ČÁST | 27 |
| 3.4 NAPÁJECÍ ČÁST | 27 |
| 3.5 ZOBRAZOVACÍ ČÁST..... | 28 |
| 3.6 PAMĚŤOVÁ ČÁST | 28 |
| 3.7 SHRNUTÍ..... | 29 |
| II PRAKTICKÁ ČÁST | 30 |
| 4 PROGRAM PRO DIAGNOSTIKU A FUNKČNOST PRVKŮ | 31 |

| | | |
|----------|--|-----------|
| 4.1 | NAHRÁVÁNÍ PROGRAMŮ | 31 |
| 4.2 | VYTVOŘENÉ STRUKTURY | 32 |
| 4.3 | VYTVOŘENÉ KNIHOVNY | 32 |
| 4.3.1 | Knihovna teploty a vlhkosti | 32 |
| 4.3.1.1 | Funkce teploty a vlhkosti | 33 |
| 4.3.2 | Knihovna LCD | 33 |
| 4.3.2.1 | Funkce zobrazení na LCD display | 34 |
| 4.3.2.2 | Funkce změny snímků | 35 |
| 4.3.3 | Knihovna Wifi | 36 |
| 4.3.4 | Knihovna web server | 37 |
| 4.3.5 | Knihovna času | 40 |
| 4.3.5.1 | Funkce určení dne | 41 |
| 4.3.5.2 | Funkce převodu data a času do EEPROM a zpět | 42 |
| 4.3.6 | Knihovna okruhu | 43 |
| 4.3.6.1 | Funkce otevření/zavření okruhu | 44 |
| 4.3.7 | Knihovna paměti | 44 |
| 4.3.7.1 | Načtení dat z paměti | 45 |
| 4.3.7.2 | Ukládání dat do paměti | 46 |
| 4.3.7.3 | Funkce nalezení nejbližšího cyklu | 47 |
| 5 | ALGORITMUS PRO AUTOMATICKÝ ZAVLAŽOVACÍ SYSTÉM | 48 |
| 5.1 | MANUÁLNÍ REŽIM | 49 |
| 5.2 | AUTOMATICKÝ REŽIM | 49 |
| 5.3 | ZOBRAZOVACÍ REŽIM | 50 |
| 6 | DOSAŽENÉ VÝSLEDKY A MOŽNÉ ROZŠÍŘENÍ MODELU | 52 |
| 6.1 | ROZŠÍŘENÍ MODELU | 52 |
| | ZÁVĚR | 54 |
| | SEZNAM POUŽITÉ LITERATURY | 56 |
| | SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK | 57 |
| | SEZNAM OBRÁZKŮ | 58 |

ÚVOD

V aktuální době horkých dnů, které se během posledních let opakovaly a nebylo jich málo, a s cílem šetřit vodu, ať už spodní vodu nebo dešťovou vodu, se čím dál víc využívá chytrého řešení automatického zavlažovacího systému. Tento systém může chytře využívat předpověď počasí a udržovat zelenou zahradu nebo znát pravidelné zavlažovací časy různých druhů zeleniny, a tak bez lidského zásahu regulovat časy závlah. V dnešní době takový systém je velmi drahý a je často vytvořen na míru každého zákazníka, který projeví zájem o takto vyspělý nástroj. Ovšem existují i méně vyspělé systémy, které umí pouze hloupě spustit závlahu v nastavený čas a tím plýtvat vodou například, když zrovna prší.

Práce je rozdělena do teoretické a praktické části. Teoretická část je dále rozdělená na tři kapitoly, kdy první kapitola popisuje používané snímače pro zavlažovací systémy a jejich cenově dostupné varianty. Druhá kapitola popisuje metody zavlažování a dostupné produkty pro automatické zavlažování na českém trhu. Poslední kapitola rozděluje zavlažovací systém na části a popisuje možné použité komponenty pro vybrané části. V závěru poslední kapitoly je souhrn použitých komponentů pro praktickou část.

Kapitoly dále pokračují v praktické části. Čtvrtá kapitola popisuje vývojové studio Arduino IDE a nahrávání projektů do řídicí desky. Popisuje vytvořené struktury a knihovny použité pro vytváření algoritmu pro automatické zavlažování. Pátá kapitola popisuje vytvořené režimy a jejich fungování. Poslední kapitola popisuje dosažené výsledky a možné rozšíření modelu.

I. TEORETICKÁ ČÁST

1 POPIS SNÍMAČŮ, AKČNÍCH ČLENŮ A ŘÍDÍCÍCH SYSTÉMŮ

Moderní zavlažovací systémy, které se umí postarat o zahradu i v nepřítomnosti majitele, vyžadují pro svoji činnost řadu různých komponentů. Náleží mezi ně snímací prvky, akční výkonové prvky a řídicí jednotka. Výsledná automatizace může být plně automatizována nebo jen částečně, kdy uživatel bude muset ve výjimečných situacích zasáhnout (1).

1.1 Snímače

Počítače jsou samy o sobě slepé. Bez snímačů nedokážou zpracovávat žádné informace. Je to to stejné, jako by se měl člověk rozhodovat, kdyby mu chyběly základní smysly jako je hmat nebo zrak. Ke každému průmyslovému počítači musí tedy být připojeny snímače – senzory, které počítači poskytují informace o stavu prostředí, které sledují a řídí (2).

Snímač je základní prvek, který je v přímém kontaktu se sledovaným prostředím a poskytuje o něm počítači informace. Je tedy zdrojem informací o fyzikálních, chemických, ale i biologických stavech v daném prostředí. Získané informace nejčastěji převádí na elektrické signály (2).

1.1.1 Snímač teploty

Pro snímání teploty se může využít mnoho různých metod pro získání požadované hodnoty. Například dilatační snímače, odporové snímače nebo termoelektrické snímače.

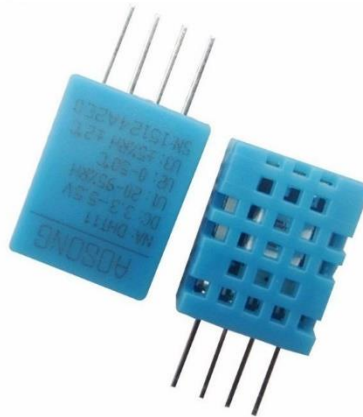
Mezi odporové snímače patří často používaný typ DS18B20, který má vodotěsné provedení a rozsah od -55 °C až do $+125\text{ °C}$, přičemž v rozsahu od -10 °C až do 85 °C má přesnost $0,5\text{ °C}$. Díky svojí konstrukci, rozsahu a přesnosti měření se snímač může využít pro měření teploty v kapalinách (3).



Obrázek 1: Snímač DS18B20 (4)

Teploměr využívá sběrnici 1 – Wire, která využívá ke komunikaci pouze jeden vodič a další dva vodiče zajišťují napájení senzoru. Napájení senzoru se může použít buď 3,3 V nebo 5 V.

Dále lze použít snímač DHT11, který nemá voděodolnou konstrukci a je určen pro měření teploty od 0 °C až do 50 °C s odchylkou 2 °C při 25 °C. Ideální pro domácí prostředí. Navíc snímač měří i vlhkost ovzduší od 20 % do 90 % RH s odchylkou při 25 °C 5 % RH (3).



Obrázek 2: Snímač DHT11 (5)

Přestože má snímač 4 vývody, tak se používají pouze 3. Na čtvrtém vývodu není žádný vstup ani výstup. Napájení snímače je možné od 3,3 V až do 5,5 V. Pro snímač jsou dostupné knihovny, které přímo zajistí získání parametrů z naměřených hodnot (3).

1.1.2 Snímač deště

Snímač deště je potřebnou součástí systému, který může i zastavit celý systém, pokud bude špatně nastavený nebo nebude komunikovat.

Snímače deště jsou mechanického původu nebo kapacitního. Kapacitní snímače se používají u automobilů pro automatické stěrače. Ovšem u kapacitních je problém určit přesně dešťové srážky a fungují jako indikátor deště. Mechanické snímače deště se využívají u větších meteostanic a umí určit i dešťové srážky.

Pro snímač pomocí kapacity lze využít desku senzoru deště s komparátorem LM393, který může být napájen 3,3 V nebo 5 V. Obsahuje logický a analogový výstup pro komunikaci s řídicí jednotkou (3).



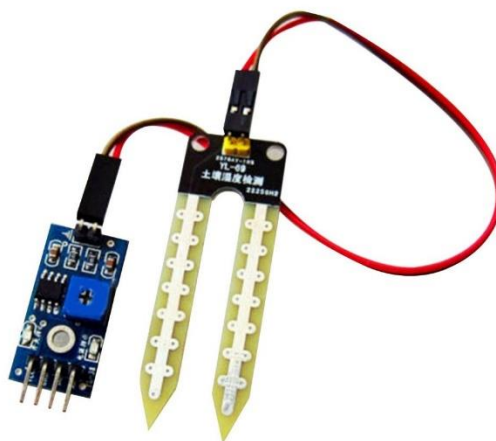
Obrázek 3: Snímač deště s komparátorem LM393 (6)

1.1.3 Snímač půdní vlhkosti

Snímač půdní vlhkosti zajišťuje informaci, jestli závlaha funguje správně. Pokud by po skončení závlahy tento snímač neukazoval očekávaná data, pak se na jeho základě může změnit další cyklus závlahy.

Snímače půdní vlhkosti se skládají z elektrody a komparátoru. Elektroda má přímý kontakt s půdou a komparátor převede naměřená data na čitelné hodnoty pro řídicí systém.

Pro tyto účely lze použít snímač skládající se ze sondy a komparátoru LM393, který může být napájen 3,3 V anebo 5 V. Dále disponuje logickým a analogovým výstupem pro komunikaci s řídicí jednotkou (3).



Obrázek 4: Snímač vlhkosti půdy a komparátor LM393 (5)

1.1.4 Snímač průtoku kapaliny

Měření průtoku patří mezi nejdůležitější měření v automatizaci. Informace bývají využívány k regulaci dávkování v technologických procesech. Kapaliny se měří většinou v uzavřeném potrubí, výjimečně v otevřeném (7).

Snímač průtoku neboli průtokoměr má doplňující informace pro celý systém závlahového systému. S jeho pomocí lze diagnostikovat poruchu na systému vodovodního potrubí. Také lze s touto informací pracovat a kontrolovat okruhy závlahy a případně celkový čas závlahy urychlit.

Průtokoměry se dají rozdělit podle metody měření. Jsou objemová měřidla, která využívají turbínu nebo lopatku, dále existují průřezová měřidla, plováčková, termoelektrická, indukční nebo ultrazvuková (7).

Na turbínovém principu je postaven průtokoměr YF-S201, který je cenově dostupný. Zvládá měřit průtok v rozsahu od 1 až po 30 l/min ovšem mezi 0 a 1 l/min je méně přesný. Snímač je možné napájet v rozsahu 5 V až 18 V a provozní teplota je $-25\text{ }^{\circ}\text{C}$ až $+80\text{ }^{\circ}\text{C}$. Pro zapojení do oběhu systému je k dispozici $\frac{1}{2}$ palcový závit. Měření průtoku probíhá detekováním pulzů na výstupu s charakteristikou 450 pulzů na litr kapaliny (3).



Obrázek 5: Průtokoměr YF-S201 (5)

Pro složitější výstup snímače se doporučuje připojit výstup na přerušeni nebo za pomoci jednočipové desky vytvořit komparátor, který by mohl posílat data o průtoku vyššímu systému.

1.2 Akční členy

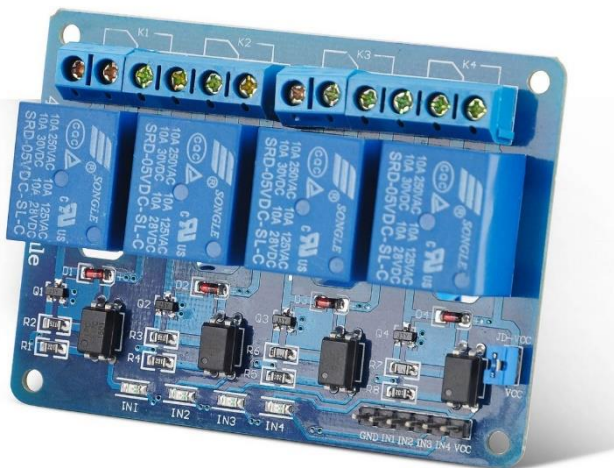
Akční člen ovládá reálnou fyzickou soustavu, kdy na vstupu dostane malý impuls a na druhé straně ovládá výkonovou jednotku. Jedná se hlavně o silovou část řízenou systémem.

1.2.1 Elektromagnetické relé

Pro nespojitý signál se využívají elektromagnetické relé, které mají pouze 2 stavy, kdy se sepnutí a rozepnutí elektrického proudu provede mechanicky. (7).

Jsou nejčastější používanou součástí, pokud nejsou v požadavku rychlé změny stavů. V případě požadavku rychlé změny stavů se místo elektromagnetického relé použije relé založené na polovodičích. Relé je několik variant lišících se velikostí, ovládacím napětím, řídicím napětím, druhem napětí a další (7).

Pro arduino se využívá s 5 V napájením a řízením, a jsou schopné ovládat 250 VAC pro maximální proud 10 A nebo 30 VDC pro maximální proud 10 A. Tyto hodnoty jsou dostatečně široké pro jakékoliv zařízení nebo spotřebič, který by byl řízen systémem (3).



Obrázek 6: Relé modul pro 4 kanály (3)

1.2.2 Elektromagnetické ventily

Elektromagnetický ventil se používá k řízení toku kapaliny v hydraulických systémech. Skládá se z cívky, pístu ze železného kovu a pružiny. Napětí na cívce bývá v rozmezí 12

VDC až 48 VDC nebo 110 VAC až 230 VAC. Ventily jsou dva druhy NO a NC. NO je v klidovém stavu otevřen a NC je v klidovém stavu zavřen (8).



Obrázek 7: Elektromagnetický ventil

1.2.3 Čerpadlo

Čerpadlo je soustrojí z pohonné jednotky a vlastního čerpadla, které dává vodě pohybovou energii. Čerpadlo slouží pro dopravu vody, kdy vodu buď nasává nebo vytlačuje.

System pro tlakování vody obsahuje ponorné čerpadlo, které zásobuje hlavní čerpadlo, které tlakuje vodu do zavlažovacího systému. Ponorné čerpadlo čerpá vodu z podzemní studny, která je považována za bezednou.



Obrázek 8: Tlakové čerpadlo 1 kW

Toto čerpadlo může být trvale připojené k elektrické síti a obsahuje vlastní vypínání systému pro ochranu, aby nešlo naprázdno a nespálilo se vynutí. Jedná se o čerpadlo o výkonu 1 kW a je napájeno 230 VAC s tlakovou nádobou nastavenou na 3,3 barů.

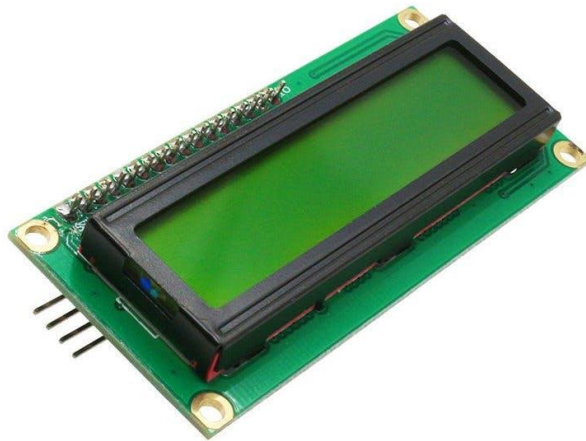
1.3 Zobrazovací členy

Pro informování uživatele je dobré zobrazovat užitečná data. Pro zobrazení je možné využít různé metody, ale všechny informují uživatele o stavu systému.

1.3.1 Zobrazení na LCD

LCD je jednoduchý zobrazovací panel, který využívá tekutých krystalů, které jsou mezi zdrojem světla a ochranným sklem displeje. Každý pixel je mezi dvěma průhlednými elektrodami a polarizačními filtry. Při minimální spotřebě se krystaly otočí o 90 stupňů a zabrání průchodu světla. Tímto způsobem se zobrazí na displeji znak (9).

Pro tyto účely se využívá displej HD44780 a pro jednoduché použití se využívá převodníku na I2C komunikaci. Displej může být 1602 nebo 2004, kdy první 2 čísla znamenají počet znaků na řádku a druhé 2 čísla počet řádků. Tyto displeje jsou průmyslovým standardem (9).



Obrázek 9: LCD displej 1602 s převodníkem na I2C (5)

1.3.2 Zobrazení OLED

Obrazovky OLED mají vyšší rozlišení než LCD displeje. Mají možnost vykreslit grafiku za pomoci více nastavitelných bodů. Ovšem s velikostí displeje roste cena. Proto moc projektů na OLED displeje není a vývojář zobrazí data na levnějším typu zobrazení v podobě znaků.

Displej je vhodný pro specifické projekty, kdy je kladen důraz na hezkou grafiku a neustálé informace o perifériích.



Obrázek 10: OLED displej s I2C komunikací (5)

1.3.3 Zobrazení pomocí webového serveru

Myšlenka tohoto přístupu zobrazení je, že každý uživatel má poblíž svůj vlastní zobrazovací displej v podobě telefonu nebo počítače. Proto je možné vytvořit webovou stránku na kterou se uživatel připojí a zobrazí si data ze systému. Webová stránka vytvořená na základě HTML5 kombinující HTML, CSS a JavaScript. V tomto případě není potřeba přidávat jakékoliv další zobrazovací panely a podoba zobrazení může být graficky výstřední.

1.3.4 Zobrazení pomocí LED diod

System může zobrazovat jednoduchá data pomocí zapnutí nebo vypnutí LED diody. Tento způsob zobrazení je velice primitivní, ale jednoznačně může říct, jaký je stav daného výstupu. Proto se tyto výstupy využívají pro oznámení nějakého stavu systému.

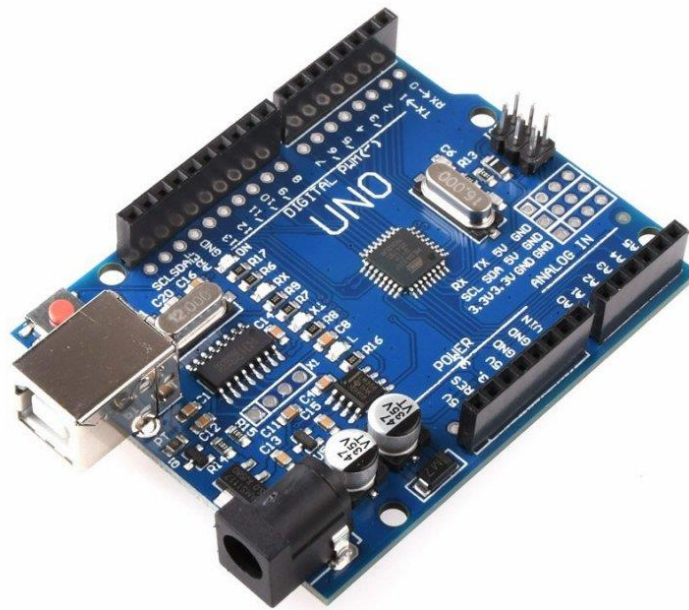
1.4 Řídicí systém

Řídicí systémy jsou programovatelná zařízení používaná nejčastěji v průmyslu, která ovládají nejčastěji technologický proces. Řídicí systém umožňuje nastavení řídicích hodnot, zobrazení informačních veličin, aktuální cyklus apod (10).

V aktuální době se řídicí systémy dostávají i do domácností, kdy lze jednoduše sestavit obvod pouze se základními znalosti z elektrotechniky za pomoci Arduino, ESP, Raspberry Pi, apod. Pro vytváření prvotních projektů je nejčastěji používána deska Arduino UNO.

1.4.1 Arduino UNO

Arduino UNO je základní deska pro prvotní vývoj projektů a je jedna z nejpoužívanějších desek na trhu v oblasti jednočipových desek. Deska obsahuje 14 vstupně/výstupních pinů, z nichž 6 pinů je možné použít jako PWM výstupy, dále má deska 6 analogových vstupů. Řídící desku je možné napájet přes USB nebo přes napájecí konektor 6-12 V (3).



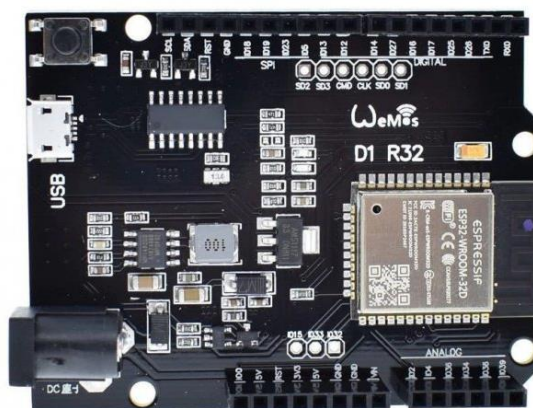
Obrázek 11: Řídící deska Arduino UNO (5)

Ale Arduino UNO neobsahuje možnost připojit se k wifi. Proto pro chytřejší projekt, pracující s moderními technologiemi, nebude možné zakomponovat všechny použité nástroje.

1.4.2 WeMos D1 R32

Řídící systém se bude skládat z WeMos D1 R32 ve verzi ESP32, která byla vydána roku 2016. Oproti dřívějšímu čipu ESP8266 je dvakrát tak rychlý, disponuje Bluetooth a obsahuje mnohem víc GPIO portů. Pro možné další rozšíření je tento řídicí modul vhodnější (3).

Modul WeMos D1 R32 s čipem verze ESP32 má velice dobrou dokumentaci hardwaru. Deska obsahuje regulátor vstupního napájení na 3,3 V a také zajišťuje sériovou komunikaci pro nahrávání programu přes microUSB kabel.



Obrázek 12: Řídící deska WeMos D1 R32 (3)

ESP32 pracuje s napětím 3,3V a výkonově je na tom o mnohem lépe než jeho starší verze ESP8266. Modul obsahuje procesor Tensulica LX6, který může dosahovat frekvencí až 240 MHz. Pro úsporný mód využívá procesor ULP. Anténa pro Wi-Fi signál je integrována na desce. Modul obsahuje i Bluetooth ve verzi 4.2 a má společnou anténu s Wi-Fi. Modul obsahuje 34 GPIO pinů, pro různé možnosti připojení periférií (5).

2 MOŽNOSTI AUTOMATIZACE ZAVLAŽOVACÍHO SYSTÉMU

Je několik možností, jak udělat automatický zavlažovací systém. Záleží na požadavku vlastníka zahrady a dispozičních možnostech. Automatizace zavlažování pak nese velké výhody pro vlastníka. Například úsporu času, kdy by musel zalívat zahradu. Dále pak úsporu peněz, kdy každá část zahrady dostane tolik vody, kolik potřebuje. Nebo možnost zalévání rostlin během pobytu na dovolené.

Pro sestavení si vlastního automatizovaného zavlažovacího systému lze využít řadu výrobců a řadu vlastních řešení. Ovšem co je nedílnou součástí je pak vyšší cena, a ta je úměrná velikosti zahrady a dispozice.

Nejlevnější řešení je vytvoření vlastního automatizovaného zavlažovacího systému, který lze sestavit za pomoci vývojové desky dle vlastního uvážení. Ale toto řešení není jednoduché a vyžaduje spoustu znalostí z elektrotechniky a programování.

2.1 Metody závlahy

Je několik možností, jak provádět zavlažování prostoru a je nutné přihlídnout k několika aspektům.

Záleží na druhu plodin, které se pěstují. Na klimatickém podnebí, kde je závlaha využívána. Na množství potřebné vody a jejich možných zdrojích. Na druhy zeminy, jestli je rovinná nebo hornatá. Na počtu měsíců v roce, kdy je potřebná závlaha rostlin. A další aspekty, které můžou přinést komplikace při výběru metody závlahy.

2.1.1 Metoda zavodňování

Metoda zavodňování je nejjednodušší formou a není potřeba používat čerpadlo. Využívá se hlavně ve svahovém terénu, kdy může voda lehce téct z jednoho konce na druhý a rovnoměrně se vstřebává do půdy. Není příliš účinná a vyžaduje velké množství vody. Při této metodě nastává problém, že se voda nedostane ke kořenům rostlinstva. Metoda je typická pro tropické podnebí.

2.1.2 Metoda postřiku

Metoda postřiku je nejrozšířenější a je využívána po celém světě. Pro správné fungování musí být dostatečný tlak v potrubí, který potřebuje každý postřikovač zvlášť. Tlak vytváří čerpadlo napojené na zdroj vody. Postřikovače jsou otočné nebo statické. Největší výhodou

jsou nízké náklady na pořízení, protože v tryskách nejsou pohyblivé části a jsou jednoduché na výrobu. Tato metoda se neobejde bez čerpadla, které tlakuje potrubí závlahy. Na trysky se dají přidělat rozprašovače a tím je zabráněno vytvářet velké kapky. Velká kapka by mohla poškodit úrodu a případně ji znečistit. Díky své povaze tato metoda není vhodná do větrných oblastí. Otáčení postřiků je způsobeno vodním tlakem a jsou aktivovány rázem nebo vnitřním ozubeným kolem. Postřikovač může být na jednom místě, nebo na pohybujícím se vozítku či zařízení. Postřikovače můžou být ve variantě pop-up, kdy jsou primárně schované pod povrchem země a při použití se vysunou za pomoci vodního tlaku. Tyto postřikovače jsou ideální na zavlažování trávníků (1).

2.1.3 Metoda kapkové závlahy

Metoda je typická pomalým dávkováním vody ke kořenům rostlin a tím mají vysokou účinnost. Díky této metodě nedochází k odpařování nebo odtoku vody. Používá se zvláště v suchých oblastech, kde je deficit vodních zdrojů. Používá se v ovocných sadech, ve vinohradech, na okrasných plochách nebo i ve sklenících. Pro tuto metodu je velice důležité využít kapkovač, který dávkuje velmi nízké množství kapek vody. Kapkovače se dají rozdělit podle toho, jestli mají samoregulaci nebo ne. Pokud je s regulací, pak zajišťuje stabilní dodávku vody k rostlině, pokud bude v rozmezí daných tlaků. Pokud je kapkovač bez samoregulace, pak je dodávka vody k rostlině závislá na daném tlaku v potrubí (1).

2.2 Možnosti na trhu

Na Českém trhu je pár firem, které nabízejí svoje zařízení pro zavlažovací systémy. Ve velké většině se jedná o miniaturní počítač, který se připojí na kohoutek a v určité intervaly spouští závlahu zahrady jako celku. Některé systémy disponují okruhy, kdy se spouští jeden po druhém nebo dle nastaveného časovače. Skoro žádný systém nedisponuje možností vzdáleného zrušení závlahy nebo možnosti sledovat, zda přišlo během dne nebo případné úpravy na závislosti okolní teplotě. Tyto vlastnosti ale můžou být zohledněny, pokud se dokoupí další zařízení, které tyto vlastnosti snímají.

Pro představení je v této práci uvedena jedna značka.

2.2.1 Gardena

Značka Gardena se na našem trhu objevuje už delší dobu a lze její produkty nalézt jak v kamenných obchodech, tak i v podobě e-shopů. Jako velkou výhodou lze uvést snadnou

instalaci, kterou zvládne každý. Na Českém trhu mají dobrou servisní podporu a na internetových stránkách jsou k nalezení i podrobné návody pro všechny jejich prodávané produkty. Nabízejí i možnost rozvrhnout si zahradu a vytvořit plán závlahy. Zavlažovací počítače jsou řešeny pomocí baterií (11).

2.2.1.1 Zavlažovací počítače

Značka Gardena nabízí několik zavlažovacích počítačů. Základem jsou zavlažovací hodiny, které zavlažují po nastavený čas. Jejich výhodou je pouze v cenové dostupnosti a lehké montáži, kdy se připojí na kohoutek a nastaví se, jak dlouho mají zavlažovat. Ale postrádají jakoukoliv další logiku, automatické spuštění a další (11).



Obrázek 13: Gardena zavlažovací hodiny (11)

Dalším produktem je zavlažovací počítač Flex, který by se už mohl nazvat řídicí jednotkou. Je určen především pro menší zavlažování na balkónech a terasách. Všechny data a nastavení zobrazuje na LC display. Zavlažovací počítač lze nastavit na časové spínání a s nastavením doby závlah. Ovšem čas závlahy lze nastavit pevně na předvolené hodiny. Spolupracuje i s půdním čidlem vlhkosti od Gardena a při dostatečně vlhké půdě se závlaha neaktivuje (11).



Obrázek 14: Gardena zavlažovací počítač Flex (11)

Dalším ale už pokročilejším řídicím systémem je zavlažovací počítač Master. Ten umožňuje individuální nastavení zavlažovacích časů a společně s příslušenstvím rozdělovače vody může zavlažovat několik okruhů postupně. Všechny data a nastavení zobrazuje na LC display a podporuje 10 jazyků. Tento zavlažovací počítač je možné kombinovat s půdním čidlem vlhkosti od značky Gardena a při dostatečně vlhké půdě se závlaha neaktivuje (11).



Obrázek 15: Gardena zavlažovací počítač Master (11)

Pokročilou technologií disponuje zavlažovací počítač Smart, který nedisponuje displejem a funguje na principu Cloudu. Umožňuje pokročilé nastavení závlahy a jeho řízení pomocí mobilní aplikace GARDENA. K počítači se musí dokoupit Gardena Smart Gateway, pro komunikaci s cloudovými službami. V aplikaci jsou všechny potřebné informace a nastavení, která jsou již přednastavená výrobcem. Úprava těchto dat je velice jednoduchá a intuitivní (11).



Obrázek 16: Gardena smart zavlažovací počítač (11)

2.2.1.2 Příslušenství Gardena

Značka Gardena disponuje spoustou příslušenství ke svým systémům závlahy. Patří k nim Smart půdní senzor, který informuje řídicí systém o vlhkosti půdy, venkovní teplotě a intenzitě světla. Řídicí systém pak dostává informace a může určit, zda se bude zavlažovat zahrada. Ovšem pro fungování vyžaduje Gardena Smart Gateway (11).



Obrázek 17: Gardena smart půdní senzor (11)

Dalším příslušenství je Rozdělovač vody automatic, který zajišťuje automatické zavlažování až šesti různých oblastí postupně. Zavlažování každé oblasti jde nastavit individuálně. Využívá se pro při nedostatku vody pro všechny koncové přístroje (11).



Obrázek 18: Gardena rozdělovač vody automatic (11)

3 NÁVRH ROZHRANÍ ŘÍDÍCIHO SYSTÉMU

Řídicí systém disponuje několika typy komunikace pro spolehlivou funkčnost, aby mohl komunikovat se senzory a získávat potřebná data. Řídicí systém komunikuje se senzorovou a akční částí. Dále je popsána napájecí část. Zavlažovací oblast je vytvořena pro 4 okruhy zavlažování. Systém obsahuje i zobrazovací část, která zobrazuje naměřená data ze senzorů a stavy okruhů z akčních členů.

3.1 Řídicí část

Pro řízení systému je vhodné vybrat desku, která má dostatek I/O pinů a případnou přidanou hodnotu. Pro rychlý vývoj jsou vhodné desky Arduino UNO a jejich klony programovatelné v Arduino IDE. Nevýhodou klonů je pak možná nekompatibilita s open source knihovnamí, které jsou dostupné pro originální desky a přidané nadstavby nemusí fungovat správně nebo vůbec.

Z více uvedených důvodů se většina projektů sestavuje na originální vývojové desce Arduino UNO. Pokud je požadavkem rozšiřující modul, jsou k dispozici podporované knihovny.

K zachování kompatibility je vhodné vyměnit některé knihovny pro komunikaci a využít jinou vývojovou desku. Jako náhrada mohou být desky od výrobce WeMos s čipem ESP. Tímto způsobem lze nahradit vývojovou desku a přidat hodnotu v podobě možnosti využít wifi modulu bez složitého programování.

3.2 Senzorová část

Mezi senzorovou část patří DHT11, který měří okolní teplotu a vlhkost. Snímač je v blízkosti řídicí desky a hlídá vnitřní teplotu řídicího systému, zda nedochází k přehřívání.

K systému může být přidán snímač deště, který zaznamenává případný déšť. Snímač měří i délku deště a na základě této události se může systém rozhodnout o odložení nebo přeskočení zavlažovacího cyklu.

Snímač deště nemusí být jediným způsobem k rozhodnutí o odložení nebo přeskočení zavlažovacího cyklu. Snímač půdní vlhkosti může stanovit, že daná oblast už závlahu nepotřebuje a tím nespustí daný cyklus. Tím nahrazuje snímač deště a řídí odložení cyklů.

Pro přesné měření venkovní teploty lze použít snímač DS18B20, který je svojí konstrukcí vhodný pro venkovní použití. Snímač měří venkovní teplotu nebo teplotu potrubí a kontroluje, zda nejsou přizemní mrazíky a nepovolí, aby se systém závlahy spustil. Pro celoroční provoz je tato funkce výhodná.

Průtokový snímač informuje systém o přesných datech o tom, kolik vody bylo spotřebováno při zavlažování. Snímač při nastavení maximálních hodnot může indikovat poruchu zavlažovacího okruhu.

3.3 Akční část

Do akční části náleží ovládání jednotlivých okruhů. Hlavní částí okruhu je tlakové čerpadlo, které tlakuje zavlažovací systém na požadovaný tlak, který je nutný pro potrubní systém a postřikovače. Čerpadlo není přímo ovládané systémem, ale má svoje vlastní ochranné prvky a tím je bezúdržbové.

Pokud máme zavlažovací systém natlakovaný, spustí se jednotlivé okruhy pomocí elektromagnetických ventilů. Ventil je v základním stavu zavřený a otevře se až po přivedení napětí. Ventily je možné řídit různým napětím, záleží na vybraných parametrech.

Pro řízení ventilů a oddělení silové části od řídicí, se využívá elektromagnetické relé. Relé je ovládané řídicím systémem na základě nastavených cyklů uživatelem. Relé poté ovládá silovou část, která spíná elektromagnetické ventily a následně otevře cestu průtoku zavlažovací kapaliny na postřikovače. Každé relé je zvlášť ovládáno vlastním vstupním pinem a komunikace probíhá pomocí booleovských stavů.

3.4 Napájecí část

Systém vyžaduje rozdělení napěťové úrovně mezi různými komponenty pro jejich napájení nebo komunikaci. Některé komponenty mají větší rozsah napětí pro řízení a napájení, a jiné fungují pouze na jednom napájecím rozsahu. Proto se musí vytvořit napájecí okruhy pro různé komponenty.

Řídicí systém je možné napájet od 5 V do 12 V. Pro dostatečný výkon je zvolen 12 V zdroj o výkonu 12 W a adaptér je napájen 230 VAC. Řídicí deska převede vstupní napájení na 5 V a na 3,3 V, které můžeme využít pro napájení ostatních částí systému.

Snímače mohou být napájeny 3,3 V nebo 5 V, které můžou být vyvedeny z řídicí desky, nebo přidáním externího zdroje. Pro menší počet komponentů lze využít napájení z řídicí

desky. Pokud by byl počet komponentů vysoký, je nutné využít externí zdroj pro stabilitu systému.

Jelikož řídicí deska využívá pro svoje periferie pouze 3,3 V logiku, je nutné převést 5 V komunikaci na 3,3 V komunikaci. K těmto účelům slouží převodník logické úrovně, který převede napětíovou logiku posílaných a přijímaných signálů. Tímto způsobem jsou ošetřeny převody napětíové logiky.

V závislosti vyžadovaného napájecího napětí vyplývajícího z katalogového listu snímače se použije dané napájení, a v případě komunikace s řídicím systémem se využije převodník logické úrovně. Takové zařízení má LCD zobrazení s komunikací I2C, kdy se signál převede z 3,3 V logiky na 5 V logiku.

Silová část ovládaná přes elektromagnetické relé je oddělená od řízení a není nutné pro převod zasahovat.

3.5 Zobrazovací část

Uživatel systému využívá zobrazovací část pro kontrolu systému a jeho správnou funkčnost a nastavení. Systém využívá zobrazení pomocí zobrazovacího panelu typu LCD 1602 a s převodníkem komunikace na I2C, která zajistí jednodušší komunikaci s řídicím systémem. Díky převodníku je do systému možné přidat další zobrazovací panel a systém v jednom okamžiku zobrazí více dat nebo je zrcadlí na jiném místě.

Systém může disponovat i webovým rozhraní, které zrcadlí data z displeje nebo přidá zobrazovací hodnotu. Následně lze pomocí webového rozhraní upravovat a přidávat zavlažovací cykly a měnit parametry systému.

3.6 Paměťová část

Aby systém po odpojení ze sítě neztratil svoje data, musí si je ukládat do uložení. Moderní systémy fungují na principu cloudu, kdy jsou data uložena v databázi na serveru, ke kterému se připojují. Dalším ze způsobů je využití vlastní paměti, kdy jsou data uložena v EEPROM paměti, která po odpojení napájení nejsou smazána. V místních sítích se využívá uložení NAS, které obsahuje potřebná data, která jsou přístupná na místní síti.

Funkce cloudu bývá z pravidla placená služba a chráněna přístupovými metodami. Ke službě se lze dostat i mimo místní síť a data jsou přístupná odkudkoliv.

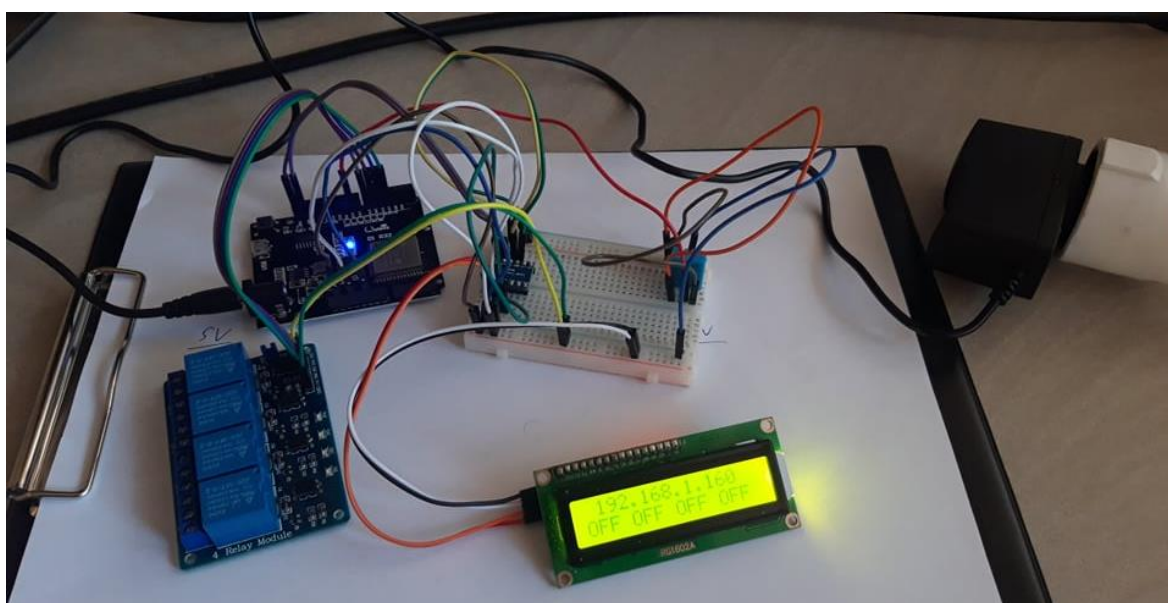
Pro místní síť bývá využit vlastní server, který disponuje zálohováním souborů nebo jejich uložením i po výpadku napájení. Vlastní server může být v podobě NAS nebo vlastního řešení, které obsahuje API pro komunikaci mezi systémy.

Nejjednodušší řešení je ukládat data do vlastní EEPROM paměti a z ní načíst data po znovu obnovení napájení. Nevýhodou paměti EEPROM je, že není stavěna pro opakované zápisy dat do buněk a časem se buňka poškodí. Poté nepůjde do buňky zapisovat.

3.7 Shrnutí

Sestavený systém obsahuje od každé části vybrané komponenty a jsou plně naprogramovány pro ucelené řízení systému.

Řídící deska je vybrána pro dostatek volných řídicích I/O pinů a možnosti wifi připojení WeMos D1 R32. Pro senzorovou část byl vybrán snímač DHT11, který měří teplotu a vlhkost v těsné blízkosti řídicí desky. Z části akčních členů je použit modul se čtyřmi elektromagnetickými relé výstupy pro oddělení silové části od řídicí. Napájení je řešeno pomocí připojeného 12 V adaptéru o výkonu 12 W, připojeno na řídicí desku a z desky vyvedeno 5 V a 3,3 V pro ostatní komponenty. K zobrazení je použito LCD 1602 se žlutým podsvícením a s převodníkem na I2C komunikaci, který zobrazuje dostupné informace ve smyčce. Pro zobrazení je využito i naprogramované webové rozhraní, které funguje i jako konfigurační závlahy. Jako paměťová část je použita EEPROM paměť, která je dostupná na řídicí desce.



Obrázek 19: Sestavený systém

II. PRAKTICKÁ ČÁST

4 PROGRAM PRO DIAGNOSTIKU A FUNKČNOST PRVKŮ

Pro sestavení funkčního celku je zapotřebí v první řadě otestovat všechny komponenty a jejich fungování v prostředí. Je zapotřebí vyzkoušet ovládání a komunikaci, která je v této práci navrhována.

V první řadě se musí otestovat fungování a nahrávání programů a podprogramu do řídicího systému, který je v této práci zvolený jako WeMos D1 R32. Deska je podobná desce Arduino UNO, ale některé vlastnosti desky jsou jiné. Podstatnou změnou je využití čipu ESP32, který je výkonnější než čip použitý v Arduino UNO a disponuje funkcí připojení k wifi nebo Bluetooth.

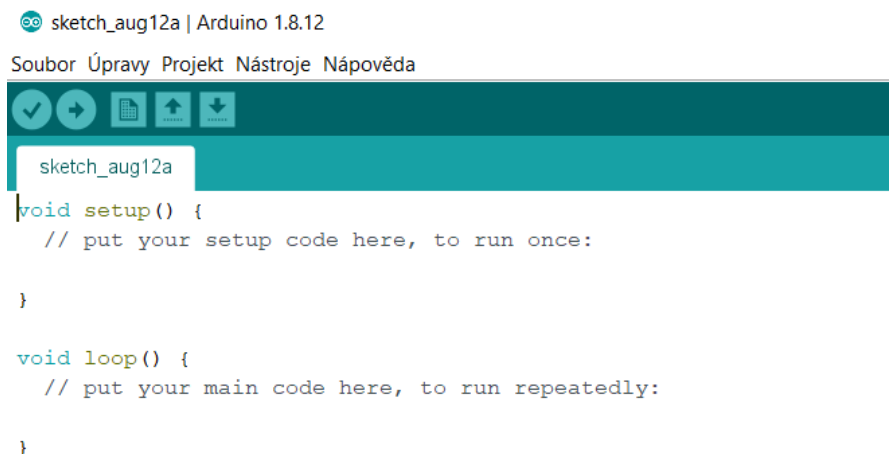
Deska není přímo v podporovaných deskách pod programem Arduino IDE a jeho knihovně dostupných desek. Proto se musí použít pro kompilaci kódu vývojová deska univerzálního označení „ESP32 Dev Module“, která spadá pod knihovnu desek Arduino ESP32 ze stránek <https://github.com/espressif/arduino-esp32>.

Po stažení a nastavení Arduino IDE je možné plnohodnotně použít tuto vývojovou desku.

4.1 Nahrávání programů

Pro nahrání programu je využito programové studio Arduino IDE, kdy je zapotřebí nahrát informace o desce a použité knihovny z internetových stránek.

Programování probíhá v jazyce Wiring, který se velmi podobá OOP C++. Napsaný program se následně nahraje přes sériovou linku do paměti čipu.



```
sketch_aug12a | Arduino 1.8.12
Soubor Úpravy Projekt Nástroje Nápověda
sketch_aug12a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Obrázek 20: Nový projekt v Arduino IDE

Základní struktura programu se skládá ze dvou částí, které musí obsahovat každý program pro Arduino.

První část s názvem *setup()* se volá po zapnutí řídicí desky a zde se provádí všechny nastavovací parametry. Jedná se o nastavení vstupně výstupních pinů, nastavení komunikace, připojení k wifi a další.

Druhá část je volána v opakujících se cyklech s názvem *loop()*. Zde se vyhodnocují všechny hodnoty, které jsou potřebné pro systém.

4.2 Vytvořené struktury

Pro ucelení potřebných dat pro automatický provoz je dobré vytvořit strukturu, která všechny informace uchovává a pod indexem je lze použít. V této práci je vytvořena struktura *cyklus*, která obsahuje informace pro spuštění automatické závlahy. Jedná se o id daného okruhu, dny spuštění, spouštěcí hodina, spouštěcí minuta a dobu běhu závlahy v minutách.

```
struct cyklus{
    int okruh;
    int dnySpusteni;
    int startHodina;
    int startMinuta;
    int dobaBehu;
};
```

Obrázek 21: Wiring struct cyklus

4.3 Vytvořené knihovny

Pro přehlednost kódu a případný refaktoring je nejlepší vytvořit si vlastní knihovny a v nich funkce, které jasně říkají svým názvem, co provádí případně jaký parametr vrací.

4.3.1 Knihovna teploty a vlhkosti

Pro snímač typu DHT11 je využita knihovna *DHT.h*, která pro použití vyžaduje zadat typ snímače a číslo pinu pro komunikaci. Poté lze začít pracovat s daným snímačem a vytvářet si vlastní funkce.


```
#include <DHT.h>

#define DHTPIN 13
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void startDHT() {
    dht.begin();
}
```

Obrázek 22: Wiring knihovna teploty a vlhkosti

4.3.1.1 Funkce teploty a vlhkosti

Po inicializaci knihovny v setup části lze použít jeho další funkce pro získání teploty a vlhkosti. Pro zpřehlednění a pozdější zkrácení kódu jsou hodnoty dostupné v číselné nebo textové podobě.

```
float getTemperature() {
    return dht.readTemperature();
}

int getHumidity() {
    return dht.readHumidity();
}

String getTemperatureString() {
    String t = String(getTemperature());
    String dataString = t.substring(0, t.indexOf(".")) + t.substring(t.indexOf("."), t.indexOf(".") + 2);
    return dataString;
}

String getHumidityString() {
    return String(getHumidity());
}
```

Obrázek 23: Wiring funkce teploty a vlhkosti

4.3.2 Knihovna LCD

Pro LCD display je vytvořena knihovna, která využívá I2C komunikaci pro zobrazování na LCD o velikosti 16 znaků a 2 řádky. I2C komunikace pro svoji funkčnost vyžaduje adresu portu. Pro display je to adresa v HEX 0x3F. Vytvořená knihovna je ošetřena pro případ, že nebude připojen LCD displej, což znamená, že se systém nebude snažit posílat data pro zobrazení. Proto při zavádění je display testován, zda je na daném portu připojen. V případě odpojení jsou globální proměnnou informované ostatní funkce, že tato možnost není dostupná. Tato ochrana je vytvořena za pomoci knihovny *Wire.h* a pro zobrazovací komunikaci je použita knihovna *LiquidCrystal_I2C.h*.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

uint8_t portLCD = 0x3F;
LiquidCrystal_I2C lcd(portLCD, 16, 2);
boolean runLCD = false;

void startLCD() {
  // Kontrola připojeného LCD na daném portu.
  Wire.begin();

  byte error;
  Wire.beginTransmission(portLCD);
  error = Wire.endTransmission();

  if(error == 0) {
    lcd.begin();
    lcd.backlight();
    runLCD = true;
    Serial.println("LCD connect");
  } else {
    runLCD = false;
    Serial.print("LCD is not connect: ");
    Serial.println(Wire.endTransmission());
  }
}
```

Obrázek 24: Wiring knihovna LCD

4.3.2.1 Funkce zobrazení na LCD display

Pro jednoduchost zobrazení textů je vytvořena funkce, která se postará o zobrazení dat na display. Pokud by byla tato funkce volána opakovaně se stejnými parametry, pak je nastavena kontrola a tento text by se nevypsál znovu na display. Tímto se zamezí blikání displeje. Zobrazená zpráva je navíc vycentrována pro intuitivnější zobrazení. Při zavádění programu je lepší, aby se po zobrazení nějakou dobu vyčkalo, než se přepne obraz a bude se pokračovat v zavádění. Proto byla přidána navíc funkce, která vypíše text, ale vyčká stanovenou dobu.

```
String copyText1 = "";
String copyText2 = "";

void showLcd(String text1, String text2) {
    if(!copyText1.equals(text1) || !copyText2.equals(text2)) {
        copyText1 = text1;
        copyText2 = text2;
        if (runLCD) {
            // Kod pro zápis na LCD displej
            lcd.clear();
            int centerText = (16 - text1.length())/2;
            lcd.setCursor(centerText,0);
            lcd.print(text1);

            centerText = (16 - text2.length())/2;
            lcd.setCursor(centerText,1);
            lcd.print(text2);
        } else {
            Serial.println("LCD isn't show text: ");
        }
        // Poslaný text je vložen do sériové linky pro debugování
        Serial.print(text1);
        Serial.print("          ");
        Serial.println(text2);
    }
}

void showLcdDelay(String text1, String text2, int timeDelay) {
    showLcd(text1, text2);
    delay(timeDelay);
    //if (runLCD) lcd.clear();
}
```

Obrázek 25: Wiring funkce zobrazení na LCD

4.3.2.2 Funkce změny snímků

Během běhu programu se přepínají snímky obrazovky po daných intervalech. Tím je docíleno zobrazení všech požadovaných parametrů během chvilky na malém display. Pokud by byl větší display, pak by se mohlo zobrazit více dat a nemusela by se přepínat obrazovka. Pro aktuálnější data je k funkci nastavena obnovovací frekvence, která nepřepne snímek, ale znovu načte data a zobrazí.

```
void setFrame() {
    switch (frameId) {
        case 0 : setIpAndStatusFrame();
                break;
        case 1 : setTempAndHumFrame();
                break;
        case 2 : setDateAndTimeFrame();
                break;
        case 3 : setNextOpenFrame();
                break;
        default: frameId = 0;
                setFrame();
                break;
    }
}

void nextFrame() {
    frameId++;
}

void changeFrame() {
    if (millis() - ttTime > changeTime) {
        nextFrame();
        ttTime = millis();
    }
    if (millis() - tttTime > frameRateTime) {
        setFrame();
        tttTime = millis();
    }
}
```

Obrázek 26: Wiring funkce změny snímku

4.3.3 Knihovna Wifi

Velkou výhodou řídicího modulu je možnost připojení k wifi, která je v blízkosti. Komunita okolo Arduina nabízí velkou škálu možností správy wifi. Od nejjednodušší, kdy se nastaví v programu SSID a password daného AP, přes vlastní vytvořené AP, které by bylo uzavřené, až po komplexní, kdy se pomocí vlastního AP nastaví na okolní AP uživatelem.

Jelikož je známé SSID a heslo sítě, ke které se bude systém připojovat, tak je zvoleno přímé připojení za pomoci SSID a hesla.

```
#include <WiFi.h>

char* ssid      = "ssid";
const char* password = "password";
boolean runWifi = false;

void startWifi() {

    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);
    byte wificNT = 0;
    while (WiFi.status() != WL_CONNECTED && wificNT < 25) {
        delay(500);
        Serial.print(".");
        wificNT++;
    }

    Serial.println("");
    if (WiFi.status() == WL_CONNECTED) {
        runWifi = true;
        Serial.println("WiFi connected");
    } else {
        runWifi = false;
        Serial.println("WiFi not connected");
    }
    Serial.println("");
}
```

Obrázek 27: Wiring knihovna wifi

4.3.4 Knihovna web server

Knihovna vytváří spojení komunikace řídicí desky a GUI pro zadávání parametrů od uživatele systémů za pomoci webové stránky. Webový server pro svoji funkčnost potřebuje nastavit port, na kterém má poslouchat příchozí zprávy. Jelikož se jedná o webovou stránku HTML a protokol HTTP, pak se nastavuje výchozí port 80.

Server funguje na principu dotazů. Pokud přijde dotaz bez parametrů, pak se zobrazí základní HTML stránka.

```
server.on("/", [] () {
    server.send_P(200, "text/html", html);
    Serial.println("Connect user");
});
```

Obrázek 28: Wiring zobrazení základní webové stránky

Ovšem stránka je vytvořena tak, že při načtení stránky se pošle dotaz na server, který odpoví s daty ve formátu JSON s jednotlivými stavy řídicího systému. Přijátá data JavaScript vyhodnotí a zobrazí je na webu.

```
server.on("/infoData", [] () {
    String data = "{\"odpoved\":1";
    data += ", \"time\": \"#time\" ";
    data += ", \"dayofweek\": \"#dayofweek\" ";
    data += ", \"temp\": \"#temp\" ";
    data += ", \"humidity\": \"#humidity\" ";

    data += ", \"stavR1\": \"#stavR1\" ";
    data += ", \"stavR2\": \"#stavR2\" ";
    data += ", \"stavR3\": \"#stavR3\" ";
    data += ", \"stavR4\": \"#stavR4\" ";

    data += "}";

    data.replace("#time", getAllTimeNowString());
    data.replace("#dayofweek", getDayNowString());
    data.replace("#temp", getTemperatureString());
    data.replace("#humidity", getHumidityString());

    data.replace("#stavR1", String(getState(1)));
    data.replace("#stavR2", String(getState(2)));
    data.replace("#stavR3", String(getState(3)));
    data.replace("#stavR4", String(getState(4)));

    server.send(200, "application/json", data);
});
```

Obrázek 29: Wiring odeslání JSON data klientovi

V případě čtení dat z uložených cyklů je vytvořen další dotaz, pro lepší rozdělení dat. Tento dotaz se odešle a zobrazí při kliknutí na tlačítko Load. Na serveru se načtou požadovaná data a ve formátu JSON se odešlou zpět na připojeného klienta a za pomoci JavaScriptu se zobrazí v tabulce.

```

server.on("/cyklyData", []() {
    int id = server.arg("idMem").toInt();
    String data = "{\\"odpoved\\":1";

    data += ",\\"idMem\\":\\"#idMem\\" ";
    data += ",\\"okruh\\":\\"#okruh\\" ";
    data += ",\\"startDays\\":\\"#startDays\\" ";
    data += ",\\"startTime\\":\\"#startTime\\" ";
    data += ",\\"runTime\\":\\"#runTime\\" ";

    cyklus cData = getData(id);

    data.replace("#idMem", String(id));
    data.replace("#okruh", String(cData.okruh));
    data.replace("#startDays", getIntToDays(cData.dnySpusteni));

    data.replace("#startTime", getStartTime(cData.startHodina, cData.startMinuta));
    data.replace("#runTime", String(cData.dobaBehu));

    data += "}";

    Serial.println(data);
    server.send(200, "application/json", data);
});

```

Obrázek 30: Wiring odeslání JSON cyklus klientovi

Při ukládání dat se odesílá na server dotaz GET s parametry. Server přijme data a uloží si je do paměti. Přijatá data následně pošle zpátky ve formátu JSON.

```

server.on("/save", []() {

    cyklus newCyklus;

    int idMem = server.arg("idMem").toInt();
    newCyklus.okruh = server.arg("okruh").toInt();
    newCyklus.dnySpusteni = getDaysToInt(server.arg("startDays"));
    newCyklus.startHodina = getHour(server.arg("startTime"));
    newCyklus.startMinuta = getMinute(server.arg("startTime"));
    newCyklus.dobaBehu = server.arg("runTime").toInt();

    saveData(idMem, newCyklus);

    String data = "{\\"odpoved\\":1";
    for(int i = 0; i < server.args(); i++) {
        data += ", \\" + server.argName(i) + \\":" + server.arg(i);
    }
    data += "}";

    Serial.println(data);

    server.send(200, "application/json", data);

});

```

Obrázek 31: Wiring uložení dat od klienta

Pokud bude uživatel chtít ovládat okruh manuálně přes GUI, pak se v podobě dotazu pošle informace o daném okruhu, který se má sepnout nebo rozepnout.

```
server.on("/rele", []() {
    int idRele = server.arg("id").toInt();
    int state = server.arg("state").equalsIgnoreCase("on");

    if (state) {
        openOkruh(idRele);
    } else {
        closeOkruh(idRele);
    }
    setManualOnOff();

    String data = "{\"odpoved\":\"1\"";
    data += "}";

    server.send(200, "application/json", data);

});
```

Obrázek 32: Wiring sepnutí/rozepnutí okruhu

Po nastavení všech dotazů a funkcí, které se mají při dotazu zavolat, se může server zapnout.

```
// Spuštění HTTP serveru
server.begin();
Serial.println("");
Serial.println("Web Server is ONLINE");
Serial.println("");
```

Obrázek 33: Wiring spuštění webového serveru

Nyní je server dostupný za pomoci TCP IP na IP adrese zařízení na portu 80.

4.3.5 Knihovna času

Pro získání času je několik možností. Pokud by byl zvolen zastaralý způsob, tak by byl připojen krystal RTC ke zvolené desce a získával z něj čas. Ovšem díky připojení k internetu je možné získat čas pomocí protokolu NTP, kterým se například řídí Windows Time.

NTP je síťový komunikační protokol pro synchronizování času mezi klienty. Jeho úkolem je synchronizovat všechny účastníky sítě do koordinovaného světového času UTC během pár milisekund.


```
// NTP
#define _NTP_PACKET_SIZE 48
#define _NTP_SERVER "pool.ntp.org"
#define _NTP_ZONE 1
#define _NTP_PORT 8888
#define _NTP_REFRESH 3600

int ntp_summer = 1;
```

Obrázek 34: Wiring základní parametry pro stažení času

Díky knihovně *TimeLib.h* lze nastavit externí čas z NTP serveru do řídicího systému a nastavit opětovnou synchronizaci systému za určitou dobu.

```
void startTime() {
    udp.begin(_NTP_PORT);
    setSyncProvider(getNtpTime);
    setSyncInterval(_NTP_REFRESH);
}
```

Obrázek 35: Wiring načtení a nastavení synchronizace času

Nyní je čas v řídicím systému a programátor s ním může jednoduše pracovat za pomoci standardních funkcí.

4.3.5.1 Funkce určení dne

Během práce s cykly je zapotřebí určit aktuální den a nastavený den. Aktuální den dostaneme z knihovny *TimeLib.h*, ale pouze v číselné podobě s tím, že začíná od neděle. V našem časovém pásmu se začíná od pondělí, a tak se musela přizpůsobit i výstupní data. Pro správné zobrazení byl vytvořen převodník mezi textovou a číselnou formou.

```
String getDayString(int dayInt) {
    switch(dayInt) {
        case 1: return "NE";
        case 2: return "PO";
        case 3: return "UT";
        case 4: return "ST";
        case 5: return "CT";
        case 6: return "PA";
        case 7: return "SO";
    }
    return "unknow";
}

int getDayInt(String dayString) {
    if (dayString.equals("NE")) return 1;
    if (dayString.equals("PO")) return 2;
    if (dayString.equals("UT")) return 3;
    if (dayString.equals("ST")) return 4;
    if (dayString.equals("CT")) return 5;
    if (dayString.equals("PA")) return 6;
    if (dayString.equals("SO")) return 7;
    return 0;
}

String getDayNowString() {
    return getDayString(weekday());
}
```

Obrázek 36: Wiring funkce určení dne

4.3.5.2 Funkce převodu data a času do EEPROM a zpět

Kvůli potřebě uložit zadané dny do paměti a načíst z paměti dny spuštění okruhu, je vytvořena funkce pro převod dat. Z webu jsou přijata data v podobě zkratky dne od uživatele a do paměti jsou ukládány v číselné podobě. Proto pro zjednodušení práce s dnem jsou vytvořené funkce pro převod dat.

```
// Převod z webu do EEPROM paměti
int getDaysToInt(String days) {
    int d = 0;
    if(days.indexOf("NE") > -1) d += 1;
    if(days.indexOf("PO") > -1) d += 2;
    if(days.indexOf("UT") > -1) d += 4;
    if(days.indexOf("ST") > -1) d += 8;
    if(days.indexOf("CT") > -1) d += 16;
    if(days.indexOf("PA") > -1) d += 32;
    if(days.indexOf("SO") > -1) d += 64;
    return d;
}
```

Obrázek 37: Wiring funkce pro převod dne do paměti

K tomuto se vztahuje i převod času zadaného od uživatele v textové formě na číselnou formu pro uložení do paměti. K této funkci je vytvořena i opačná funkce, která převádí data zpátky z paměti k uživatelskému rozhraní.

```
// Práce pro server - převod časových formátů od webu do čipu a práce s EEPROM pamětí
String getStartTime(int h, int m) {
    String ho, mi;
    if (h < 10) {
        ho = "0" + String(h);
    } else {
        ho = String(h);
    }
    if (m < 10) {
        mi = "0" + String(m);
    } else {
        mi = String(m);
    }
    return ho + ":" + mi;
}

int getHour(String t) {
    if (t.indexOf(":") > -1) {
        return t.substring(0, t.indexOf(":")).toInt();
    } else {
        return 0;
    }
}

int getMinute(String t) {
    if (t.indexOf(":") > -1) {
        return t.substring(t.indexOf(":") + 1, t.indexOf(":") + 3).toInt();
    } else {
        return 0;
    }
}
```

Obrázek 38: Wiring funkce pro převod času mezi uživatelem a pamětí

4.3.6 Knihovna okruhu

Pro ovládání jednotlivých okruhů přes relé obvody, je vyčleněn jednotlivý pin pro ovládání. Tento pin se musí inicializovat jako výstupní pin a nastavit jeho počáteční poloha. Relé modul je sepnut, pokud je nastavena hodnota LOW, a rozepnut, pokud je nastavena hodnota HIGH. Jinak je na modulu použit pull-up rezistor, pro zajištění klidového napětí na stav rozepnuto.

```
#define OKRUH_1 14
#define OKRUH_2 27
#define OKRUH_3 16
#define OKRUH_4 17

void closeAll();

void initOkruh() {
    pinMode(OKRUH_1, OUTPUT);
    pinMode(OKRUH_2, OUTPUT);
    pinMode(OKRUH_3, OUTPUT);
    pinMode(OKRUH_4, OUTPUT);
    closeAll();
}
```

Obrázek 39: Wiring definování a nastavení výstupů pro ovládání okruhů

4.3.6.1 Funkce otevření/zavření okruhu

Pro sepnutí nebo rozepnutí relé je nutné zadat id relé, které se má sepnout případně rozepnout. Následně se daný okruh sepne případně rozezne.

```
void openOkruh(int id) {
    Serial.println("Open okruh " + String(id));
    switch(id) {
        case 1: digitalWrite(OKRUH_1, LOW);
            break;
        case 2: digitalWrite(OKRUH_2, LOW);
            break;
        case 3: digitalWrite(OKRUH_3, LOW);
            break;
        case 4: digitalWrite(OKRUH_4, LOW);
            break;
        default:
            break;
    }
}
```

Obrázek 40: Wiring funkce pro otevření okruhu

4.3.7 Knihovna paměti

Aby byl systém bezproblémový po případném vypnutí a zapnutí systému z jakéhokoliv důvodu (výpadek napájení, restartování systému) a uživatel nemusel nastavovat parametry

závlahy ručně, je vytvořena knihovna paměti EEPROM pro ukládání nebo načítání zadaných parametrů uživatelem.

Při spuštění systému se knihovna inicializuje a načte uložená data z paměti v dané struktuře. Jedna buňka paměti obsahuje 8 bitů, čímž lze dostat hodnoty od 0 po 255. Kvůli složitějšímu způsobu alokování paměti je zvoleno statické místo dynamického. Toto je možné si dovolit, protože je na desce pouze jedna aplikace využívající prostředky systému.

```
#define _MAX_CYKL 20
#define DATA_LENHT 5
#define _EEPROM_SIZE 512

cyklus cykly[_MAX_CYKL];

void startMemory() {
    EEPROM.begin(_EEPROM_SIZE);
}
```

Obrázek 41: Wiring inicializace paměti

4.3.7.1 Načtení dat z paměti

Při spuštění systému se musí načíst data z paměti EEPROM do paměti RAM. Čtení z EEPROM je mnohem pomalejší než z RAM paměti. Pro práci s daty je mnohem výhodnější je mít načtené v RAM paměti a změny případně ukládat do EEPROM paměti.

```
cyklus loadData(int id) {
    int okruh = EEPROM.read(id * DATA_LENHT);
    int startDays = EEPROM.read((id * DATA_LENHT) + 1);
    int startHour = EEPROM.read((id * DATA_LENHT) + 2);
    int startMinute = EEPROM.read((id * DATA_LENHT) + 3);
    int runTime = EEPROM.read((id * DATA_LENHT) + 4);

    cyklus ret;

    ret.okruh = okruh;
    ret.dnySpusteni = startDays;
    ret.startHodina = startHour;
    ret.startMinuta = startMinute;
    ret.dobaBehu = runTime;

    return ret;
}

void loadAllData() {
    for(int i = 0; i < _MAX_CYKL; i++) {
        cykly[i] = loadData(i);
    }
}
```

Obrázek 42: Wiring funkce načtení paměti do systému

4.3.7.2 Ukládání dat do paměti

Aby bylo možné načíst data, musí je uživatel nejprve nastavit a uložit. Při ukládání dat může nastat, že buňka paměti je poškozená a data nebyla uložena. V tomto případě uživatel musí využije jiné adresy.

```
// Vrací 0, pokud úspěšně uložil hodnotu
int saveVal(int address, int val) {
    EEPROM.write(address, val);
    EEPROM.commit();
    int testVal = EEPROM.read(address);
    if(val != testVal)
        return 1;
    else
        return 0;
}
```

Obrázek 43: Wiring funkce pro ukládání dat

4.3.7.3 Funkce nalezení nejbližšího cyklu

Pro pokročilou možnost zobrazení, které může zobrazit následující cyklus, je vytvořena speciální funkce, která projde všechny cykly a vrátí cyklus, který je nejbliž k aktuálnímu času. Pro získání potřebných dat jsou vytvořené 3 podobné funkce, které vypíší den spuštění, okruh pro otevření a čas spuštění.

```
String getNextCyklDay() {
    String dIndex = getDayNowString();
    String retDIndex = "0";
    int c = 0;
    while(c < 7) {
        int sH = 24;
        int sM = 60;
        for(int i = 0; i < _MAX_CYKL; i++) {
            String dDays = getIntToDays(cykly[i].dnySpusteni);
            if (dDays.indexOf(dIndex) > -1) {
                if (cykly[i].startHodina > hour()) {
                    c = 10;
                    if (sH - hour() > cykly[i].startHodina - hour()) {
                        sH = cykly[i].startHodina;
                        sM = cykly[i].startMinuta;
                        retDIndex = dIndex;
                    }
                    if (sH - hour() == cykly[i].startHodina - hour()) {
                        if (sM - minute() > cykly[i].startMinuta - minute()) {
                            sH = cykly[i].startHodina;
                            sM = cykly[i].startMinuta;
                            retDIndex = dIndex;
                        }
                    }
                }
            }
            if (cykly[i].startHodina == hour()) {
                if (cykly[i].startMinuta > minute()) {
                    c = 10;
                    if (sM - minute() > cykly[i].startMinuta - minute()) {
                        sH = cykly[i].startHodina;
                        sM = cykly[i].startMinuta;
                        retDIndex = dIndex;
                    }
                }
            }
        }
        dIndex = nextDay(dIndex);
        c++;
    }
    return retDIndex;
}
```

Obrázek 44: Wiring funkce získání následujícího spuštění

5 ALGORITMUS PRO AUTOMATICKÝ ZAVLAŽOVACÍ SYSTÉM

Pro sestavení algoritmu je zapotřebí sestavit webové rozhraní pro ovládání a zadávání parametrů pro zavlažování. Webové rozhraní ovládá i ruční ovládání pro spínání relé. Jsou zobrazeny čtyři ovládací relé, které jdou zapnout a vypnout.

Zavlažovací Systém

| Relé 1 | Relé 2 | Relé 3 | Relé 4 |
|---------|---------|---------|---------|
| Vypnuto | Vypnuto | Vypnuto | Vypnuto |
| Zapnout | Zapnout | Zapnout | Zapnout |
| Vypnout | Vypnout | Vypnout | Vypnout |

Aktuální čas: 19:06:35
 Aktuální den: Pondělí
 Teplota: 24.2°C
 Vlhkost: 59%
 Info: Nothing

| Dny spuštění | | | | | | | Okruh | Čas spuštění | Čas vypnutí | | |
|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-----------|--------------|-------------|------|------|
| PO | ÚT | ST | ČT | PÁ | SO | NE | | | | | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | Okruh 3 ▾ | 21 : 47 🕒 | 1 | Save | Load |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | Okruh 2 ▾ | 08 : 30 🕒 | 10 | Save | Load |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | Okruh 2 ▾ | 08 : 45 🕒 | 10 | Save | Load |
| <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Okruh 1 ▾ | 10 : 42 🕒 | 2 | Save | Load |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Okruh 1 ▾ | 21 : 17 🕒 | 2 | Save | Load |

Obrázek 45: Webová podoba stránky systému

Každé relé náleží pro jeden zavlažovací okruh. Webový prohlížeč následně zobrazí nový stav daného relé pro okruh. Webová stránka ukazuje aktuální čas, který má stažený z NTP serveru a aktuální den, který je taktéž stažený z NTP serveru, který tuto informaci poskytuje. Webová stránka zobrazuje teplotu a vlhkost naměřenou v blízkosti řídicího systému. Tato informace může být poskytnuta vyššímu řídicímu systému.

Dále je zobrazena tabulka, do které může uživatel zadat časy, dny, okruh a dobu spuštění závlahy. Po stisknutí tlačítka Save se uloží data do paměti zavlažovacího systému. Při prvním otevření webové stránky se data automaticky nenačtou do tabulky kvůli velkému

zatížení systému při odeslání úvodní stránky. Proto je vyžadováno jednotlivé načtení dat u každého cyklu zvlášť pomocí tlačítka Load.

5.1 Manuální režim

Ruční ovládání okruhů je dostupné z webové stránky. Manuální režim pomocí tlačítek není implementován v této verzi. Manuální režim se nevypne automaticky, a proto musí uživatel při skončení používání režimu vypnout všechny otevřené okruhy.

5.2 Automatický režim

Automatický režim funguje neustále pokud jsou zadané cykly pro spuštění, systém je pod napětím a připojený k místní síti pomocí wifi z důvodu aktuálního času v systému. Systém během každého vlastního cyklu kontroluje, zda nenastal čas pro sepnutí relé a otevření okruhu.

```
int sTime = 0;
int sCircle = 0;

boolean isThisTimeOpen() {
    for(int i = 0; i < _MAX_CYKL; i++) {
        String d = getIntToDays(cykly[i].dnySpusteni);
        //Serial.println(getDataCykluString(cykly[i]));
        if (d.indexOf(getDayNowString()) > -1) {
            if (cykly[i].startHodina == hour()) {
                if (cykly[i].startMinuta == minute()) {
                    if (second() < 2) {
                        sTime = millis() + cykly[i].dobaBehu*60*1000;
                        sCircle = cykly[i].okruh;
                        return true;
                    }
                }
            }
        }
    }
    return false;
}
```

Obrázek 46: Wiring funkce pro automatický režim

Pokud bude nastavený čas a aktuální čas shodný, vrátí hlavní smyčce hodnotu TRUE a do globálních proměnných nastaví okruh pro zavření a dobu za jakou se má zavřít. Hlavní smyčka poté nedovolí otevřít druhý okruh, dokud se tento okruh nezavře.

```
if ((millis() - tTime > controlTime) && (stopTime == 0)) {
    tTime = millis();

    if (isThisTimeOpen()) {
        thenOpen();
        stopTime = getStopTime();
        closeCircle = getCircle();
    }
}
```

Obrázek 47: Wiring hlavní smyčka řízení otevření okruhů

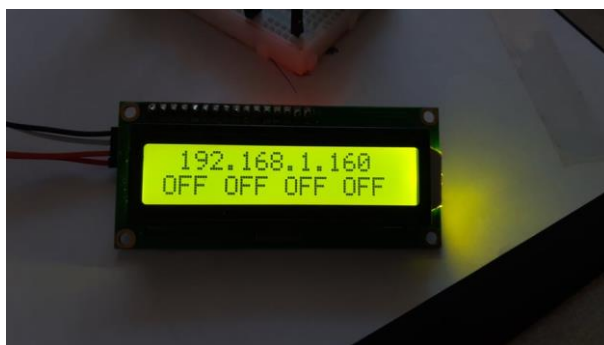
Během hlavní smyčky se kontroluje, zda neuběhla nastavená doba pro uzavření smyčky. Pokud nastala, uzavře daný okruh, který byl naposledy otevřen na základě zadaných parametrů uživatelem.

```
if (millis() > stopTime && stopTime != 0) {
    closeOkruh(closeCircle);
    stopTime = 0;
}
```

Obrázek 48: Wiring hlavní smyčka řízení zavření okruhu

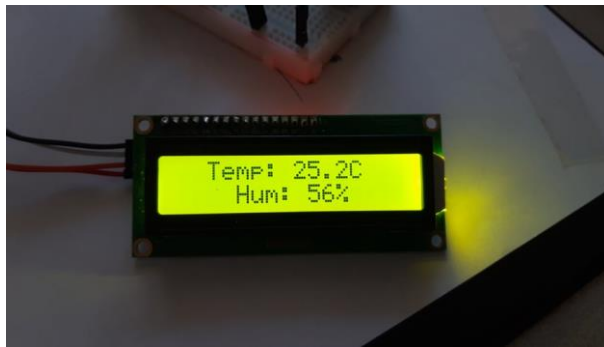
5.3 Zobrazovací režim

Zobrazovací režim běží nezávisle na automatickém nebo manuálním režimu. Na LCD displeji se přepínají v daných intervalech informační obrazce a každou vteřinu se obnovuje změna.



Obrázek 49: Zobrazovací panel – IP adresa a stavy okruhů

5.3Obrázek 49 zobrazuje IP adresu a stav jednotlivých okruhů. Pokud bude okruh sepnut, do vteřiny se obnoví obrazec a informuje pozorovatele o změně.



Obrázek 50: Zobrazovací panel – teplota a vlhkost systému

Obrázek 50 zobrazuje teplotu a vlhkost z DHT11 v blízkosti řídicí jednotky. Pozorovatel tak vidí aktuální hodnoty v okolí řídicí jednotky.



Obrázek 51: Zobrazovací panel – aktuální den a čas

Obrázek 51 zobrazuje aktuální čas a den ve zkratce. Každou vteřinu se obnovuje zobrazení, takže aktualizace času může mít prodlevu v rámci milisekund.



Obrázek 52: Zobrazovací panel – Následující otevření okruhu s dnem a časem

Obrázek 52 zobrazuje čas a den dalšího otevření okruhu. Pokud nebude nastavené další spuštění, zobrazí se nulové hodnoty.

6 DOSAŽENÉ VÝSLEDKY A MOŽNÉ ROZŠÍŘENÍ MODELU

System sestavený na řídicí desce WeMos D1 R32 je plně funkční a obsahuje vybrané komponenty pro otestování celku systému. Program obsahuje komunikaci pro snímač DHT11 a základní funkce pro použití v ostatních částech programu. Dále disponuje ovládací částí pro ovládání relé obvodů, které jsou připojené k daným okruhům pomocí elektromagnetických ventilů. Je naprogramováno kompletní ovládání zobrazovacího panelu, které je využito k cyklickému zobrazování dat pro dostupné informace o systému. System disponuje wifi připojením, které je programově zakomponováno a nastaveno k připojení k místní síti. Díky připojení k internetu je možné získávat aktuální čas a řešit časově nastavitelné úlohy. Dále je možné spravovat konfiguraci zavlažování za pomoci webového serveru, který zobrazuje aktuální stavy systému a nastavené parametry. System je odolný proti ztrátě paměti při odpojení z napájení díky vytvoření vlastní struktury ukládání dat a její implementace do systému.

System se skládá ze 2 částí, ze zaváděcí části a cyklické části. V zaváděcí části se načte a nastaví kompletní system podprogramů, aby mohly fungovat všechny vybrané periferie systému. V cyklické části se zpracovávají požadavky připojeného klienta přes webový server, kontroluje se aktuální čas s časy nastavených cyklů pro spuštění a ukončení cyklů zavlažování a provádějí se změny zobrazovacího panelu. Všechny hodnoty komponentů se aktualizují v případě dotazu na ně.

6.1 Rozšíření modelu

Naprogramovaný system obsahuje pouze část plně dostupných komponentů.

System je možné rozšířit o další prvky jako je přesnější měření venkovní teploty, která by mohla řídit odklady nastavených cyklů na pozdější spuštění v případě, že by venkovní teplota byla příliš vysoká a zavlažování by mohlo poškodit úrodu. Navíc v horkých dnech se voda rychleji odpařuje z půdy a doporučuje se zavlažovat v ranních nebo večerních hodinách. Bez této ochrany určuje bezpečnost pouze bezpečné uživatelské nastavení času.

System lze doplnit o snímač deště, který by indikoval déšť před zavlažovacím cyklem a tím by mohl zablokovat závlahu a povolit ji následující den nebo s odstupem stanovených hodin.

Pro měření spotřebované vody během závlahy lze system vybavit průtokoměrem kapaliny, který by informoval řídicí system o spotřebované vodě na daný okruh podle uživatelského nastavení cyklů. Pro regulování průtoku by system bral v úvahu, kolik vody může

vyprodukovat na daný okruh a tím vytvořit omezení, kdy by se zavlažování automaticky vypnulo a neplývaly se dostupné zásoby vody pro zavlažování.

V rámci správné závlahy je dobré měřit vlhkost půdy v zavlažovací oblasti. Proto by se k zavlažovacímu modelu mohly přidat další modely, které by mohly být připojeny k místní wifi síti a posílaly by informace řídicímu systému, který by si je ukládal a zobrazoval. Na základě naměřených dat by mohl uživatel nebo automatická úprava zvolit delší nebo kratší časy závlah pro daný okruh.

Model je možné rozšířit přidáním ovládacích prvků v podobě tlačítek sepnutí okruhu a přepínače módu automat/manuál a diodových kontrolky o sepnutých okruzích. Dále programově upravit a nastavit základní stavy při změně módu. Přepínač módu by mohl být zakomponován i do webové podoby.

Jelikož je systém připojen k internetu, je možné nastavit notifikace uživateli, pokud je mimo místní síť a informovat ho o spuštění nebo odložení závlahy z daných důvodů. Případně jej informovat, že může nastavit lepší zavlažovací čas. K tomuto rozšíření by náleželo vytvořit i aplikaci pro mobilní platformy a vytvořit cloudovou službu, která je nad rámec této práce.

Pro vytvoření modulárního systému by bylo zapotřebí vytvořit vlastní DPS, která by obsahovala všechny komunikační moduly pro připojení a jejich vlastní napájení. DPS by mohlo i rozčlenit napájecí části a k systému by mohl být připojen pouze 12 V napájecí zdroj o dostatečném výkonu, který by zvládl napájet všechny komponenty. Deska by mohla disponovat konektory pro připojení displeje, snímače teploty, průtokoměr a připojení relé modulu nebo přímo je obsahovat na desce. DPS by mohla obsahovat i patičky pro daný řídicí systém, případně přidat další řídicí desku s vlastní komunikací mezi sebou.

ZÁVĚR

Cílem práce bylo se seznámit a popsat snímače, akčními členy a řídicí systémy, které by mohly být použitelné v automatickém zavlažovacím systému. Jsou popsány dostupné komponenty, které jsou cenově dostupné a použitelné pro vývoj uceleného systému.

Náplní práce byla analýza způsobů zavlažování a dostupných možností automatických zavlažovacích systémů na trhu. Nabízené systémy zavlažování jsou uzavřené a nepodporují přidání produktů od ostatních výrobců nebo produktů vlastní výroby.

Úkolem práce bylo navrhnout systém automatické závlahy, který využívá cenově dostupné komponenty a je nastavitelný uživatelem systému. Systém je rozdělen na několik částí, kde je každá část popsána, a které komponenty v nich lze použít. Na konci kapitoly je konečné sestavení systému, které je v praktické části naprogramováno a otestováno. Návrh systému klade důraz na nízkou pořizovací cenu a dostupnost na českém trhu.

Praktická část popisuje nahrávání projektu do řídicí desky, vytvořené struktury a knihovny pro práci s vybranými komponenty. Kapitola popisuje důležité funkce daných knihoven, které jsou využity v algoritmu automatického systému.

Díky dobře rozpracovaným knihovnám bylo konečný algoritmus snadné sestavit a vytvořit podmínky cyklů a kompletní program pro systém zavlažování. Pro účely testování jsou vytvořené 3 režimy, které fungují nezávisle na sobě.

Dosažené výsledky systému jsou popsány v poslední kapitole, která sepisuje použité komponenty a popisuje funkcionalitu systému. Kapitola popisuje i možné rozšíření systému, které by vyžadovalo i programovou úpravu.

Během práce bylo testováno použití dešťového snímače s převodníkem LM393 a kvůli rušení okolních vlivů od spínání elektromagnetického relé a blízkého napájecího vodiče nebylo možné přesně určit stav, kdy by na snímač přšelo. Stejný problém se objevil i u snímačů půdní vlhkosti se stejným převodníkem. Nejspíš byl problém i v délce komunikačního drátu a špatného odrušení. Snímače by nemusely mít v případě lepšího připojení na řídicí desku tak velké chybové problémy. Proto jsou snímače navrhnuty v rozšířeném modelu, kdy by měly svoji vlastní řídicí jednotku, která by komunikovala se systémem.

V případě ostrého provozu je důležité nastavit výchozí nastavení při přechodu mezi automatickým a manuálním režimem. Ovšem pro lepší ovládání systému uživatelem u řídicího systému je dobré využít rozšíření ovládacími prvky a přepínači. Webová stránka je vytvořena

velice staticky a načítání stránky je pomalé. Pro testovací účely je dostatečná, ale v případě ostré verze by měla být přepsána do dynamické podoby.

Práce může být podmětem pro sestavení komerčního produktu, který by byl modulárně rozšiřitelný. Pro tyto účely by bylo zapotřebí navrhnout DPS, která by zjednodušila spojování komponentů a pouze modulárně by se přidaly další periferie systému.

SEZNAM POUŽITÉ LITERATURY

- (1) VEVERKA, Vladimír. *Speciální mechanizace - závlahová technika pro zahradnictví*. 1. V Brně: Mendelova zemědělská a lesnická univerzita, 2003. ISBN 80-715-7738-3.
- (2) KOVÁŘ, Josef. *Snímání technických veličin*. Zlín, 2019.
- (3) *Drátek* [online]. ČR [cit. 2021-05-14]. Dostupné z: <https://dratek.cz/>
- (4) MICRO ROBOTICS LTD. Temperature Sensor - DS18B20. In: *Micro Robotics* [online]. 2021 [cit. 2021-05-16]. Dostupné z: <https://www.robotics.org.za/18B20-1M>
- (5) *Laskarduino* [online]. ČR [cit. 2021-05-14]. Dostupné z: <https://www.laskarduino.cz/>
- (6) *RPishop.cz* [online]. ČR: RPishop.cz, 2021 [cit. 2021-05-16]. Dostupné z: <https://rpishop.cz/>
- (7) HRUŠKA, František. *Technické prostředky automatizace IV: (snímače, převodníky, regulátory, průmyslová výpočetní technika, ovládací jednotky)*. Vyd. 3. Ve Zlíně: Univerzita Tomáše Bati, 2005. ISBN 80-731-8274-2.
- (8) *What are solenoid valves?* [online]. WTWH Media LLC, 2017 [cit. 2020-07-29]. Dostupné z: <https://www.pneumatictips.com/what-are-solenoid-valves/>
- (9) PREDKO, Michael. *Digitální elektronika bez předchozích znalostí*. 1. Brno: Computer Press, 2008. ISBN 978-80-251-2124-5.
- (10) BABJAK, Ján. *Řídicí systémy servisních robotů: Control systems of service robots*. Ostrava: VŠB - Technická univerzita, 2013. ISBN 978-80-248-3123-7.
- (11) *Gardena* [online]. [cit. 2021-05-14]. Dostupné z: <https://www.gardena.com/cz/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

| | |
|--------|---|
| °C | Stupeň Celsia |
| V | Volt |
| A | Ampér |
| W | Watt |
| VAC | Střídavé napětí |
| VDC | Stejnoseměrné napětí |
| RH | Relativní vlhkost |
| l/min | Litr za minutu |
| NO | Normal Open |
| NC | Normal Close |
| I/O | Vstupně/Výstupní |
| IP | Internet Protocol |
| LCD | Liquid Crystal Display |
| I2C | Inter – Integrated Circuit |
| OLED | Organic Light-Emitting Diode |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| LED | Light-Emitting Diode |
| USB | Universal Serial Bus |
| EEPROM | Electrically Erasable Programmable Read-Only Memory |
| NAS | Network Attached Storage |
| OOP | Object-Oriented Programming |
| HEX | Hexadecimal |

SEZNAM OBRÁZKŮ

| | |
|--|----|
| Obrázek 1: Snímač DS18B20 (4) | 11 |
| Obrázek 2: Snímač DHT11 (5)..... | 12 |
| Obrázek 3: Snímač deště s komparátorem LM393 (6) | 13 |
| Obrázek 4: Snímač vlhkosti půdy a komparátor LM393 (5) | 13 |
| Obrázek 5: Průtokoměr YF-S201 (5)..... | 14 |
| Obrázek 6: Relé modul pro 4 kanály (3)..... | 15 |
| Obrázek 7: Elektromagnetický ventil | 16 |
| Obrázek 8: Tlakové čerpadlo 1 kW | 16 |
| Obrázek 9: LCD displej 1602 s převodníkem na I2C (5)..... | 17 |
| Obrázek 10: OLED displej s I2C komunikací (5) | 18 |
| Obrázek 11: Řídící deska Arduino UNO (5) | 19 |
| Obrázek 12: Řídící deska WeMos D1 R32 (3) | 20 |
| Obrázek 13: Gardena zavlažovací hodiny (11)..... | 23 |
| Obrázek 14: Gardena zavlažovací počítač Flex (11) | 23 |
| Obrázek 15: Gardena zavlažovací počítač Master (11) | 24 |
| Obrázek 16: Gardena smart zavlažovací počítač (11) | 24 |
| Obrázek 17: Gardena smart půdní senzor (11) | 25 |
| Obrázek 18: Gardena rozdělovač vody automatic (11) | 25 |
| Obrázek 19: Sestavený systém | 29 |
| Obrázek 20: Nový projekt v Arduino IDE..... | 31 |
| Obrázek 21: Wiring struct cyklus | 32 |
| Obrázek 22: Wiring knihovna teploty a vlhkosti..... | 33 |
| Obrázek 23: Wiring funkce teploty a vlhkosti..... | 33 |
| Obrázek 24: Wiring knihovna LCD..... | 34 |
| Obrázek 25: Wiring funkce zobrazení na LCD | 35 |
| Obrázek 26: Wiring funkce změny snímku | 36 |
| Obrázek 27: Wiring knihovna wifi | 37 |
| Obrázek 28: Wiring zobrazení základní webové stránky | 38 |
| Obrázek 29: Wiring odeslání JSON data klientovi..... | 38 |
| Obrázek 30: Wiring odeslání JSON cyklus klientovi..... | 39 |
| Obrázek 31: Wiring uložení dat od klienta | 39 |
| Obrázek 32: Wiring sepnutí/rozepnutí okruhu | 40 |

| | |
|--|----|
| Obrázek 33: Wiring spuštění webového serveru | 40 |
| Obrázek 34: Wiring základní parametry pro stažení času | 41 |
| Obrázek 35: Wiring načtení a nastavení synchronizace času | 41 |
| Obrázek 36: Wiring funkce určení dne | 42 |
| Obrázek 37: Wiring funkce pro převod dne do paměti | 42 |
| Obrázek 38: Wiring funkce pro převod času mezi uživatelem a pamětí | 43 |
| Obrázek 39: Wiring definování a nastavení výstupů pro ovládání okruhů | 44 |
| Obrázek 40: Wiring funkce pro otevření okruhu | 44 |
| Obrázek 41: Wiring inicializace paměti | 45 |
| Obrázek 42: Wiring funkce načtení paměti do systému | 46 |
| Obrázek 43: Wiring funkce pro ukládání dat | 46 |
| Obrázek 44: Wiring funkce získání následujícího spuštění | 47 |
| Obrázek 45: Webová podoba stránky systému | 48 |
| Obrázek 46: Wiring funkce pro automatický režim | 49 |
| Obrázek 47: Wiring hlavní smyčka řízení otevření okruhů | 50 |
| Obrázek 48: Wiring hlavní smyčka řízení zavření okruhu | 50 |
| Obrázek 49: Zobrazovací panel – IP adresa a stavy okruhů | 50 |
| Obrázek 50: Zobrazovací panel – teplota a vlhkost systému | 51 |
| Obrázek 51: Zobrazovací panel – aktuální den a čas | 51 |
| Obrázek 52: Zobrazovací panel – Následující otevření okruhu s dnem a časem | 51 |

SEZNAM PŘÍLOH

PI CD

PŘÍLOHA P I: CD

Obsah přiloženého CD:

- Bakalářská práce v elektronické podobě
- Soubor napsaného programu a knihoven