

Jihočeská univerzita v Českých Budějovicích

Přírodovědecká fakulta

System automatizace pro komunitní zahradu prostřednictvím Arduino

Bakalářská práce

Radim Sejk

školitel: PhDr. Milan Novák, Ph.D.

České Budějovice 2016

Bibliografické údaje

Sejk Radim, 2016: Systém automatizace pro komunitní zahrady prostřednictvím Arduino

[Automation system for community gardens with Arduino. Bc.. Thesis, in Czech.] – 43p.

Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Bakalářská práce se zaměřuje na vytvoření a nasazení systému pro automatické řízení komunitních zahrad. V teoretické části se zabývá analýzou celého problému komunitních zahrad a následně možnostmi jejich automatizovaného řízení.

V praktické části je sestaven přípravek realizující automatizaci komunitní zahrady, je pro něj napsán obslužný software a vytvořeno uživatelské webové rozhraní.

Klíčová slova

Automatizace, komunitní zahrady, Arduino, webové rozhraní

Annotation

This thesis aims at the development and the realization of a fully automated community garden control system. Theoretical section analysis the overall community garden problematic followed-up with possibilities of community garden automated control.

Practical section consists of the construction of a tool that implements the community garden automation, it's service software and a web interface.

Keywords

Automation, community gardens, Arduino, web interface

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 20.4.2016

.....

Radim Sejk

Poděkování

Děkuji doc. RNDr. Marii Šmilauerové, Ph.D za konzultace ohledně pěstování rostlin a PhDr. Milanu Novákovi, Ph.D. za vedení práce a spolupráci při stavbě demonstračního přípravku.

Obsah

1. Úvod.....	7
2. Cíle práce	8
3. Metodika práce.....	8
4. Teoretická východiska práce.....	8
4.1. Problematika komunitních zahrad	9
4.1.1. Umístění	9
4.1.2. Nároky na provoz.....	9
4.1.3. Možnosti správy komunitních zahrad	10
4.2. Analýza existujících řešení.....	10
4.2.1. CLICK and GROW	11
4.2.2. GROVE	12
4.2.3. Analýza – shrnutí	14
5. Návrh obecného systému automatizace komunitních zahrad	14
5.1. Požadované vlastnosti.....	15
6. Praktická část	16
6.1. Návrh automatizačního systému komunitních zahrad.....	16
6.1.2. Konektivita	17
6.1.3. Čidlo vlhkosti a teploty vzduchu.....	17
6.1.4. Čidlo vlhkosti půdy	18
6.1.5. Display a I ² C převodník	18
6.1.6. RTC	19
6.1.7. Relé.....	20
6.1.8. Zavlažování	20
6.1.9. Detekce hladiny v nádrži.....	20
6.1.10. LED žárovky.....	20
6.1.11. LED driver	21

6.2.	Solární napájení	21
6.2.1.	Solární panel.....	22
6.2.2.	Akumulátor.....	22
6.2.3.	Regulátor	22
6.2.4.	Senzory pro měření napětí.....	22
6.3.	Software.....	22
6.3.1.	MQTT.....	23
6.3.2.	Node-RED	24
6.4.	Arduino.....	24
6.4.1.	Setup.....	25
6.4.2.	HW initialisation	25
6.4.3.	WiFi connection	25
6.4.4.	Load from EEPROM.....	26
6.4.5.	Loop	26
6.4.6.	Funkce Light	26
6.4.7.	Funkce Pump.....	26
6.4.8.	MQTT.....	27
6.4.9.	MQTT reconnect	27
6.4.10.	MQTT loop.....	28
6.4.11.	Show on LCD a Print on terminal	28
6.5.	Raspberry Pi	28
6.5.1.	Instalace operačního systému.....	28
6.5.2.	Změna hesla.....	29
6.5.3.	SSH přístup	29
6.5.4.	Instalace LAMP.....	29
6.5.5.	Mosquitto	29
6.5.6.	Node-RED	31

7.	Výsledky	33
7.1.	Demonstrační prototyp	33
7.2.	Webové rozhraní.....	34
8.	Testování.....	35
8.1.	Řešené problémy s HW	35
8.1.1.	Arduino webserver	35
8.1.2.	WiFiShield	35
8.1.3.	Kalibrace senzorů napětí	36
8.1.4.	Kalibrace senzorů vlhkosti půdy	36
8.1.5.	Opotřebení kontaktů senzorů vlhkosti.....	36
8.1.6.	Nedostatečný výkon solárního panelu.....	36
8.2.	Řešené problémy v Arduinu	37
8.2.1.	Přerušené spojení.....	37
8.2.2.	Synchronizace času pomocí NTP.....	37
8.2.3.	Kontrola času v daném intervalu.....	37
8.2.4.	Šifrování MQTT spojení	38
9.	Závěr	38
10.	Seznam literatury.....	39
11.	Seznam příloh.....	41

1. Úvod

Během hospodářské krize na konci 19. století nabídla města chudým obyvatelům možnost pěstovat si ovoce a zeleninu na veřejných pozemcích, čímž umožnila vznik prvním komunitním zahradám [1].

V současné době se lidé opět ke komunitním zahradám vrací, ať už z důvodu ekologie (pěstování plodin na lokální úrovni) nebo jako k možnosti trávení volného času. V ideálním případě si místní komunita zajistí prostor, který je vhodný pro pěstování rostlin (dostatek světla, přísun vody, ale i jednoduchý přístup). Bohužel ne vždy lze takovéto prostory zajistit.

Dalším aspektem, který se musí zohlednit, je čas. Zdaleka ne každý, kdo si chce něco vypěstovat, má v dnešní uspěchané době dostatek času na péči o komunitní zahradu.

Tato práce se zabývá možnostmi automatizace pro komunitní zahrady s využitím vývojové platformy Arduino.

2. Cíle práce

V teoretické části budou vysvětleny některé pojmy této problematiky a uvedeny možnosti, jak usnadnit a urychlit péči o takovouto zahradu pomocí elektroniky, a v praktické části bude sestaven funkční elektronický přípravek, který toto dokáže splnit.

Tento základní cíl je poměrně rozsáhlý, proto je rozdělen do několika dílčích cílů:

1. Provedení analýzy potřeb v oblasti komunitních zahrad se zaměřením na jejich automatizovanou správu.
2. Analýza dostupných řešení tohoto problému s ohledem na použité komponenty, software a celkovou použitelnost.
3. Na základě předchozího cíle (2) bude realizován prototyp řídicího systému zajišťující automatizaci komunitní zahrady.

3. Metodika práce

Pro dosažení uvedených cílů (2. Cíle práce) bude provedena analýza formou rešeršního šetření současného stavu problematiky. Analýza se bude zaměřovat na konkrétní existující systémy, s ohledem na následující vlastnosti a funkce:

1. Bezobslužná správa zahrady
2. Mobilní provedení
3. Použité komponenty
4. Celková použitelnost

Výsledkem bude zjištění funkcionalit nalezených a analyzovaných prostředků komunitních zahrad. Tyto společné prvky pak budou zohledněny v návrhu a realizaci prototypu řídicího systému komunitních zahrad.

4. Teoretická východiska práce

Při analýze a návrhu praktické části vychází tato práce z různých informačních zdrojů. Teoretická východiska práce se týkají zejména definic pojmů, uvedení do problematiky komunitních zahrad a analýz dostupných řešení těchto problémů.

4.1. Problematika komunitních zahrad

4.1.1. Umístění

Umístění komunitních zahrad v závislosti na rostlinách, které zde chceme pěstovat, je zásadní problém, který je nutno vyřešit. K tomuto problému lze přistupovat dvěma způsoby:

1. Je k dispozici místo, kde rostliny pěstovat a podle místních podmínek se zvolí konkrétní rostliny, které v tomto místě budou prosperovat.
2. Je v plánu pěstovat určitý druh rostlin a pak je tedy nutné jim vhodné podmínky vytvořit.

Na začátku procesu by se měly pozorováním zjistit přírodní podmínky dané lokality pro budoucí minimalizování údržby. Podmínkami jsou složení a kvalita půdy, povětrnostní podmínky, zastínění, hydrologické podmínky a funkční podmínky. Důležitou roli hraje i odvodnění pozemku, ohrožení v zimních obdobích mrazem, množství spadáných srážek apod. Členitost terénu a zejména existence žlabů může pomáhat nechtěné koncentraci vody či sněhu. V rámci zjištěných informací by se měl sestavit plán složení rostlin a jejich umístění z důvodů zoptimalizování podmínek [1].

Je tedy velmi důležité prostor vybírat s ohledem na tyto aspekty.

4.1.2. Nároky na provoz

Zahrady, ať už klasické, nebo komunitní, mohou být v závislosti na lokalitě relativně náročné na provoz. Tyto nároky se dají rozdělit na:

1. Energetické nároky
2. Lidské zdroje
3. Finanční zdroje

Mezi energetické nároky komunitních zahrad patří zejména zdroje vody, světla, živin, tepla a v neposlední řadě zdroje zahradnických potřeb. Pokud výběrem geografického umístění zahrady nejsme schopni rostlinám zajistit vhodné podmínky, je nutné najít jiný způsob náhrady, ať už jde o dopravu vody, udržení teplot, nebo přísvit.

Dalším provozním nárokem je potřebný čas lidí, kteří se o zahradu musí starat, aby prosperovala.

Předchozí dva body přímo úměrně souvisejí s finančními nároky na dopravu a zakoupení nutných energií, potřeb, ale i jako náhrada za čas strávený prací v komunitní zahradě.

4.1.3. Možnosti správy komunitních zahrad

K řešení výše uvedených (4.1. Problematika komunitních zahrad) problémů lze přistupovat obecně třemi způsoby:

1. Vytvoření soběstačné zahrady
2. Využití automatizace
3. Použití lidských zdrojů

Způsob vytvoření soběstačné zahrady spočívá v kombinaci správně zvoleného prostředí a vhodných rostlin. Tímto způsobem lze vytvořit naprosto soběstačnou komunitní zahradu, která kromě případné sklizně nevyžaduje žádné další zásahy. Ne vždy je však možné tento způsob využít, kvůli náročnosti na výběr zvolené lokality a požadovaných rostlin.

V případě, že není možné postupovat dle prvního způsobu, lze problémy vyřešit použitím dostupných automatizačních technologií, které ušetří většinu potřebných lidských zdrojů v podobě času stráveném při práci v komunitní zahradě. Pro toto řešení je nutné provést analýzu a návrh systému s ohledem na kvalitu a typ použitých komponent.

Poslední možností je komunitní zahrada vyžadující správu a kontrolu činností využívající lidské zdroje (pravidelné zalévání, hnojení, atd.). Případně je možné v různých modifikacích kombinovat předchozí způsoby s ohledem na danou lokalitu a potřeby rostlin.

4.2. Analýza existujících řešení

V provedeném šetření bylo nalezeno několik existujících dostupných řešení, které se zabývají automatizací komunitních zahrad. Některé z nich jsou ovšem ryze komerční a nelze o nich zjistit bližší informace o použitých technologiích, nebo jejich funkčnosti.

4.2.1. CLICK and GROW

Prvním analyzovaným produktem je CLICK and GROW (Obr. 1). Je to na obsluhu velmi nenáročný a některými prvky velmi zajímavý systém. Svými malými rozměry je vhodný spíše pro pěstování kuchyňských bylin a malých okrasných rostlin. Ve své horní části má sice LED zdroj světla, ale i sám výrobce doporučuje umístění u zdroje přirozeného světla pro lepší a kvalitnější růst.

V jeho spodní části se nachází nádržka na vodu se systémem okysličování, který zajišťuje dostatečný přísun živin rovnou ke kořenům rostlin. Nejzajímavějším prvkem tohoto systému je distribuce hotových kapslí, obsahujících semena různých bylin nebo okrasných rostlin včetně půdy a potřebných živin. CLICK and GROW se může nastavit pomocí mobilní aplikace, existující pouze pro systém iOS, jiné platformy nejsou podporovány. Nastavení je velmi jednoduché ale omezené na výběr pěstované rostliny, což je v kombinaci s hotovými kapslemi vhodné pro začínající a nezkušené pěstitele [9].

CLICK and GROW	
klady	zápory
jednoduché použití	rozměry
	málo detailní nastavení
	neumožňuje škálovatelnost
	slabá intenzita osvětlení

Tab. 1 - CLICK and GROW



Obr. 1 - CLICK and GROW [14]

4.2.2. GROVE

Druhým analyzovaným produktem je GROVE (Obr. 2). Tento produkt je založen na elektronické platformě Particle, která je přímo navržena pro použití s IoT. Celý produkt je velmi sofistikovaným ekosystémem pro růst rostlin. Ve spodní části se nalézá funkční akvárium s kompletním vybavením pro spokojený život rybiček. Tyto rybičky svými výměškami tvoří hnojivo, které se společně s vodou dostává do obou pater stojanu přímo ke kořenům zde vysazených rostlin. Voda se přes kořeny přefiltruje, živiny se spotřebují rostlinami a čistá voda se pak opět vrací zpět do akvária.

V horní části každého patra je pak LED zdroj světla. V spodním patře je slabší zdroj o příkonu 25 W, který je vhodný pro vyklíčení semen a růst kuchyňských bylin. V horním patře je pak silný zdroj o maximálním příkonu 200 W s proměnlivou teplotou barvy 2200 K –5000 K a vysokým barevným podáním CRI > 85. Dále se zde nacházejí čidla teploty a vlhkosti vzduchu a teploty vody a výšky vodní hladiny. Systém má mobilní aplikaci pro uživatelské nastavení [10].

GROVE	
klady	zápory
kompletní ekosystém	málo detailní nastavení
rozměry	škálovatelnost
kvalitní provedení	

Tab. 2 - Grove



Obr. 2 - Grove [15]

4.2.3. Analýza – shrnutí

Na základě provedeného šetření a rozboru analyzovaných komerčních řešení automatizace komunitních zahrad byly zjištěny výsledky (Tab. 3), ze kterých vychází následující kapitola (5. Návrh obecného systému automatizace komunitních zahrad).

Celkový přehled		
Vlastnost	Grove	CLICK and GROW
Škálovatelnost	NE	NE
Mobilní aplikace	ANO	ANO (pouze iOS)
Webové rozhraní	NE	NE
Detailní nastavení	NE	NE
Zavlažování	ANO	ANO
Osvit	ANO	ANO
Individuální výhoda	kompletní ekosystém	okysličování vody
Individuální nevýhoda	pořizovací cena	slabé osvětlení

Tab. 3 - Celkový přehled

5. Návrh obecného systému automatizace komunitních zahrad

Při analýze hotových produktů a po diskuzi se zkušenými pěstiteli rostlin i s potenciálními zákazníky s různými úrovněmi zkušeností plyne, že je vhodné navrhnout nový systém tak, aby odstranil zejména tyto nedostatky:

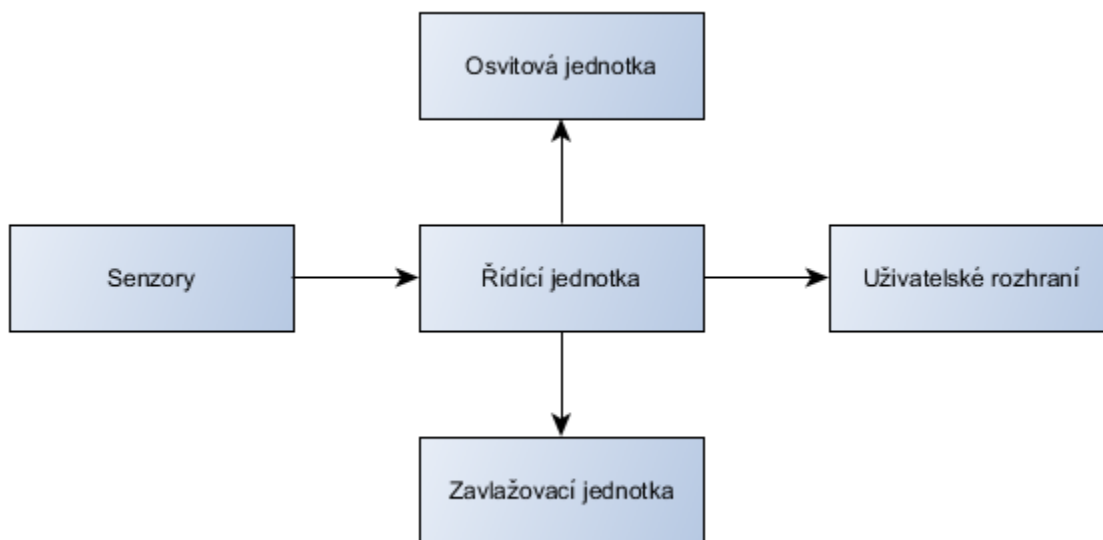
1. Nutnost manuálního zalévání
2. Nedostatečné osvětlení
3. Nedostatečně detailní nastavení systému
4. Absence možnosti škálovat celý systém

Odstranění těchto nedostatků povede k usnadnění pěstování rostlin v různých podmínkách a úspoře nutných nároků na provoz.

5.1. Požadované vlastnosti

K řešení problémů zjištěných z provedené analýzy budou využity jednotlivé prvky systému automatizace (Obr. 3). Tyto prvky a jejich vlastnosti jsou:

1. Osvitová jednotka sloužící jako zdroj světelného záření pro rostliny.
2. Zavlažovací jednotka, která bude poskytovat přísun vody.
3. Uživatelské rozhraní pro nastavení používaných veličin a zobrazení stavu celé zahrady uživateli formou lokálního rozhraní a pomocí vzdáleného přístupu. Rozhraní by mělo být rozděleno na monitorovací část a část uživatelského nastavení. Mělo by být také dostatečně detailní, aby bylo možné zajistit vhodné podmínky různě náročným rostlinám.
4. Senzorická část sloužící pro monitoring fyzikálních veličin.
5. Řídící jednotka starající se o správný chod všech předchozích částí. Automatizační algoritmus by měl mít možnost automatických i manuálně nastavitelných režimů pro pokrytí veškerých potřeb uživatele.



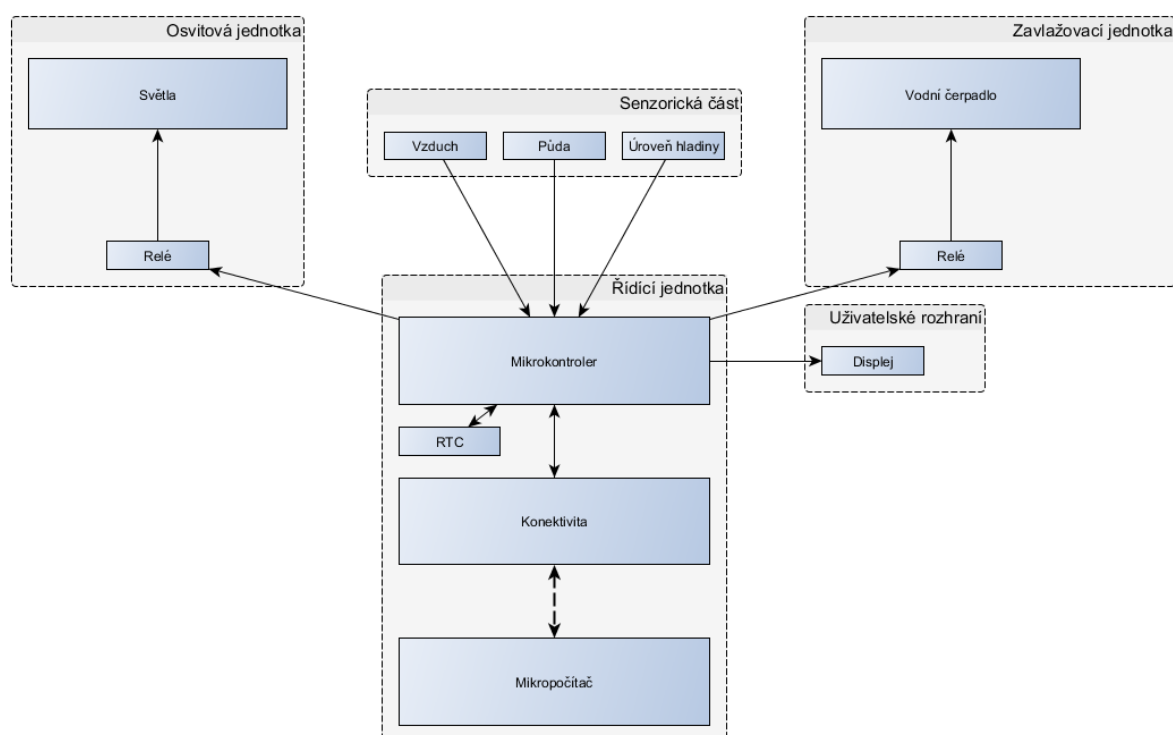
Obr. 3 - Blokové schéma funkčnosti

6. Praktická část

Pro samotné praktické řešení systému automatizace komunitních zahrad se vychází z provedené analýzy existujících produktů a obecného návrhu systému (5. Návrh obecného systému automatizace komunitních zahrad).

6.1. Návrh automatizačního systému komunitních zahrad

Zohledníme-li uvedený obecný návrh systému (5. Návrh obecného systému automatizace komunitních zahrad), tak pro konkrétní návrh lze vycházet z následujícího blokového schématu:



Obr. 4 - Blokové schéma

6.1.1. Řídicí jednotka

Řídicí jednotka by měla splňovat následující požadavky:

1. Dostatečný výpočetní výkon pro zadané operace
2. Vhodná rozhraní pro připojení vstupních a výstupních prvků

S ohledem na výše zmíněné požadavky a po provedení analýzy dostupných řešení bylo v této práci přistoupeno ke kombinaci dvou prvků. Prvním je mikrokontroler, který je vhodný pro komunikaci a připojení vstupních a výstupních částí. Druhým je server, který bude plnit roli webového rozhraní.

Jako mikrokontroler bylo zvoleno Arduino MEGA 2560. Deska je založena na mikroprocesoru Atmega 2560. Poskytuje 54 digitálních vstupů/výstupů přičemž 15 z nich může být použito pro hardwarové generování PWM signálu. Dále obsahuje 16 vstupů s A/D převodníkem pro detekci analogových veličin a 4 hardwarové linky UART. K dispozici je zde také 16 MHz krystal a Atmega 16U2 pracující jako převodník USB – sériová linka. Mikroprocesor má vlastní paměť (256 KB FLASH a 8 KB SRAM) [6].

Právě tento model byl vybrán hlavně kvůli jeho větší paměti s ohledem na celkovou náročnost zdrojového kódu a Arduino jako platforma je vhodná na tento typ použití hlavně díky široké základně návodů a propracovaných manuálů.

Pro účely demonstračního přípravku by bylo zbytečně nákladné (finančně i časově) zřizovat webhostingový server, který bude hrát roli webového rozhraní. Místo toho je použita platforma mikropočítače Raspberry Pi model B 2, která byla vyvinuta jako relativně levný prostředek k výuce a navíc je k dispozici ve škole.

Model B 2 má dostatečný výkon pro simultánní běh serveru komunikačního protokolu MQTT a zároveň i nástroje Node-RED (viz 6.3. Software).

6.1.2. Konektivita

Pro komunikaci Arduina s Raspberry Pi byla použita technologie WiFi, kterou zprostředkovává tzv. WiFi shield, čili rozšiřující deska s WiFi modulem, přímo připojitelná k Arduinu.

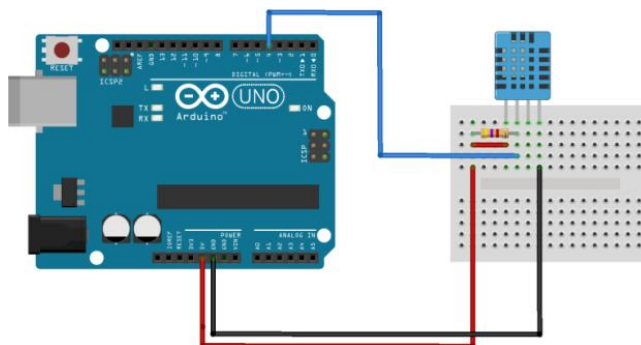
Arduino WiFi shield podporuje síť 802.11b/g, šifrování WEP a WPA2 a s Arduinem komunikuje pomocí SPI rozhraní. Je schopný zajistit TCP i UDP komunikaci a jeho součástí je i knihovna pro Arduino, která usnadňuje jeho použití a zpřístupňuje veškeré funkce [12].

6.1.3. Čidlo vlhkosti a teploty vzduchu

Jako nejvhodnější čidlo pro měření teploty a vlhkosti vzduchu se jeví AM2302, které disponuje I²C sběrnici, senzorem pro měření teploty a senzorem pro měření vlhkosti vzduchu.

Pro potřeby této práce má dostačující přesnost (Teplota: $\pm 0,5$ °C / Vlhkost: ± 3 %) a rozlišení (Teplota: 0.1 °C / Vlhkost: 0.1 %) [7].

Čidlo je v demonstračním přípravku použito pro informativní zobrazení teploty a vlhkosti vzduchu. Připojení datového pinu senzoru AM2302 je nutné provést přes pull-up rezistor (Obr. 5). V konkrétním případě demonstračního přípravku to je pin D41.



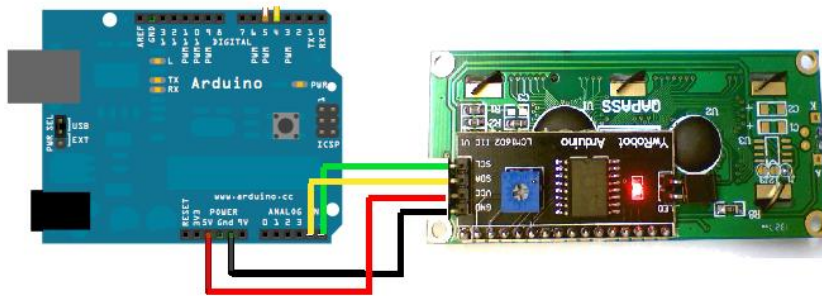
Obr. 5 - Zapojení AM2302 [13]

6.1.4. Čidlo vlhkosti půdy

K měření vlhkosti půdy byl zvolen Moisture Sensor od firmy ITEAD STUDIO. Tento senzor pomocí operačního zesilovače SGM358 měří vodivost půdy mezi jeho elektrodami a vrací analogovou hodnotu zesílenou na rozmezí 0 V až 5 V, která odpovídá vodivosti, potažmo vlhkosti půdy 0 % až 100 %, nebo digitální hodnotu s prahem nastaveným trimrem přímo na senzoru [8].

6.1.5. Display a I²C převodník

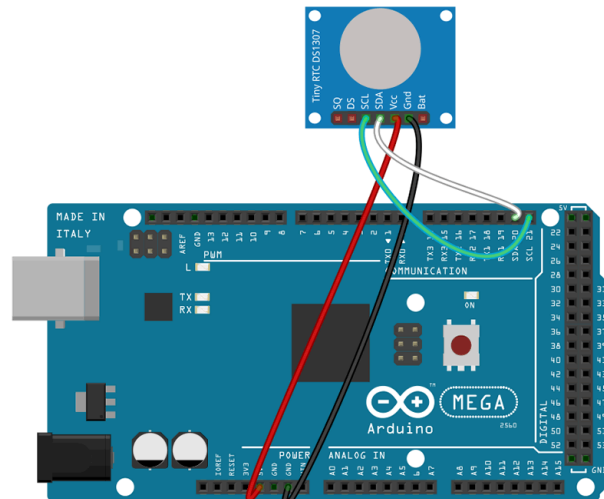
K zobrazení aktuálních nastavení a hodnot přímo v místě komunitní zahrady je použit LCD display kombinovaný s I²C převodníkem z důvodu jednodušší instalace a propojení s Arduinem. Samotný LCD displej je osazen 1602a driverem a má velikost 2x16 znaků. Jeho připojení k mikrokontroleru je realizováno dle schématu (Obr. 6).



Obr. 6 - Zapojení LCD [16]

6.1.6. RTC

Pro spínání světel a účely logování je důležité udržet v systému přesný čas. Tuto funkci splňuje RTC modul DS1307, který má dokonce vlastní záložní baterii, takže je schopný udržet čas i při odpojení systému od napájení nebo při výpadku elektrického proudu. Komunikace s Arduinem je zajištěna pomocí I²C. Připojení k mikrokontroleru je realizováno dle schématu (Obr. 7).



Obr. 7 - zapojení RTC obvodu [17]

6.1.7. Relé

Pro spínání zavlažovacího čerpadla a osvitových LED žárovek byl použit relé modul s vestavěným oddělovacím optočlenem. Optočlen je zde k odfiltrování rušivých signálů tvořených na cívce relé a také jako bezpečnostní prvek pro galvanické oddělení vysokého napětí od desky mikroprocesoru.

6.1.8. Zavlažování

Pro udržování vlhkosti v jednotlivých částech komunitní zahrady byla zvolena kombinace nádrže na vodu a ponorného čerpadla. Použité čerpadlo je německé značky Comet, typ Ocean. Je to model používaný v obytných vozech, kde je dbáno na kvalitu provedení a celkovou životnost. Tento model je určen pro napájení 12 V a má průtok 22 litrů za minutu a tlak 1,5 baru.

6.1.9. Detekce hladiny v nádrži

Pro zjištění hladiny v nádrži by bylo možné použít více typů čidel. Od ultrazvukových a infračervených detektorů vzdálenosti až po odporové snímání. Pro účely demonstračního přípravku byla ale použita plováková čidla. Malé plováky umístěné v různých výškách nádrže obsahují magnet, který spíná jazýčkový kontakt uvnitř. Je to řešení naprosto spolehlivé, levné a hlavně vodotěsné. Tato čidla jsou v nádrži umístěna tři, což je dostatečný počet pro rozdělení úrovně hladiny do 4 sekcí (Plná, skoro plná, poloprázdná a prázdná).

6.1.10. LED žárovky

Jako osvitová jednotka byly zvoleny LED žárovky. Po seznámení se s dostupnými typy LED s barevným spektrem vhodným pro růst rostlin a po konzultaci se zkušenými pěstiteli, kteří používají umělého přisvitu, byly vybrány LED žárovky GrowLight G19 Globe.

Tyto žárovky mají příkon 6 W, jsou konstruované pro napájecí napětí 12V a obsahují dvě LED – modrou s vlnovou délkou 460 až 475 nm a červenou s vlnovou délkou 620 až 735 nm. Jejich vyzařovací úhel je 180°, jsou vybaveny hliníkovým žebrováním pro lepší chlazení a

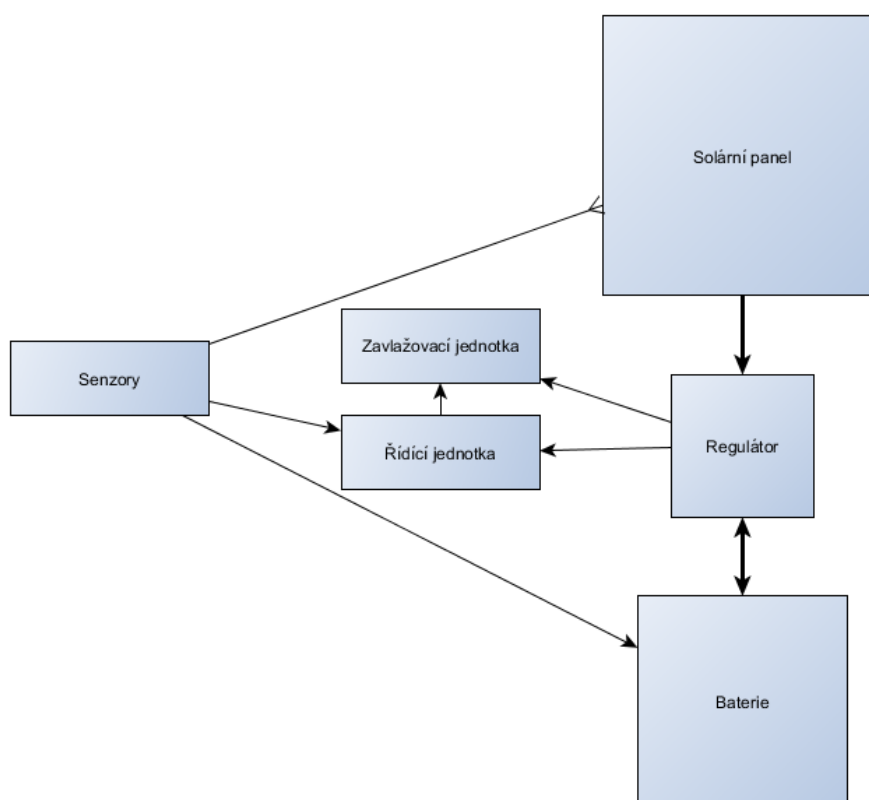
paticí MR16 pro připojení k napájecímu zdroji. Při stavbě demonstračního přípravku byly použity čtyři kusy o celkovém příkonu 24 W.

6.1.11. LED driver

Jelikož žárovky LED jsou velmi náročné na kvalitní napájení, je nutné zároveň s nimi použít vhodný driver. Proto byl použit driver MEAN WELL, typ LPF-40D-12, který je napájen z elektrické sítě 230 V a na výstupu produkuje 12 V DC o maximálním proudu 3,34 A. Tento driver navíc umožňuje regulaci výstupního výkonu a je tedy možné osvětlení stmívat na základě signálu PWM na k tomu určenému vstupu. Bohužel pořízené LED žárovky nepodporují tento typ regulace, a proto je tedy driver používán pouze v režimu zapnut/vypnut.

6.2. Solární napájení

Přestože analyzovaná řešení solárním napájením nedisponují, zde se tato možnost zcela nabízí. Použitím tohoto druhu napájení je totiž možné vyřešit problematiku napájení komunitních zahrad v místech bez zavedené elektrické sítě. Tento systém se skládá z jednotlivých prvků zapojených podle následujícího obrázku:



Obr. 8 - Blokové schéma solárního napájení

6.2.1. Solární panel

Solární panel byl použit o rozměrech 550x540x30 mm a maximálním výkonu 40 W. Jeho výstupní napětí při zátěži je až 17,82 V a až 21,96 V naprázdno.

6.2.2. Akumulátor

Použitý akumulátor je olověný, bezúdržbový typ o kapacitě 77 Ah. Tato kapacita by měla umožnit napájení celého systému přibližně dva dny (při dvanácti hodinách přisvitu denně).

6.2.3. Regulátor

Pro řízení správného nabíjecího cyklu akumulátoru a pro regulaci výstupního napětí ze solárního panelu je nutné použít solární regulátor.

Konkrétně použitý regulátor je typ CD od firmy Carspa. Tento se zapojí mezi panel a akumulátor. Případné spotřebiče se pak zapojují do speciálního výstupu z regulátoru, který je chráněn proti zkratu, odpojí se při přílišném vybití akumulátoru a nebo naopak akumulátor odpojí od nabíjení po jeho dobití. Zároveň má na sobě indikaci chybného zapojení a stavu nabití akumulátoru. Je dimenzován na maximální vstupní výkon 140 W [18].

6.2.4. Senzory pro měření napětí

Aby bylo možné sledovat stav napětí akumulátoru a velikost napětí dodávaného ze solárního panelu, bylo nutné použít senzory napětí, připojené k Arduino. Arduino jako takové je vybaveno A/D převodníkem, takže je schopné měřit i analogovou veličinu, avšak pouze do úrovně napětí 5V. Tento senzor je tvořen jednoduchým odporovým děličem napětí, který vyšší vstupní napětí (až 15 V na akumulátoru a až 22 V na solárním panelu) rozdělí na nižší napětí na výstupu (maximálně 5 V pro Arduino) v poměru podle použitých rezistorů.

6.3. Software

V této části bude popsán použitý software, jeho instalace a konfigurace. V závěru také bude popsána funkčnost a logika celé aplikace zajištěná právě tímto softwarem.

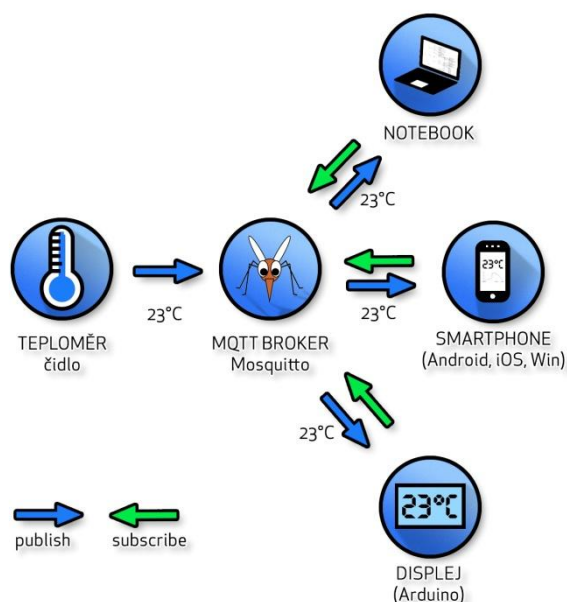
6.3.1. MQTT

MQTT je zkratka Message Queuing Telemetry Transport. Je to na data nenáročný, extrémně jednoduchý protokol, určený pro zařízení s nízkým výpočetním výkonem a pro nespolehlivé sítě s vysokou latencí a nízkou propustností. Pracuje na principu zasílání krátkých zpráv typu publish/subscribe skrze server (broker) až ke klientům (subscribers) jak je vidět na schématu (Obr. 9). Díky výše zmíněným vlastnostem je tento protokol ideálním pro M2M nebo také IoT komunikaci ve světě navzájem propojených zařízení a pro mobilní aplikace, kde rychlost přenosu dat a energetická náročnost na baterii je velmi podstatná.

MQTT vynalezl Dr. Andy Stanford-Clark z IBM a Arlen Nipper z Arcom (nyní Eurotech) v roce 1999.

MQTT standardně používá rezervovaný TCP/IP port 1883. Dále je možné použít port 8883 pro potřeby SSL.

MQTT podporuje základní autentizaci pomocí jména a hesla od verze 3.1. Šifrování komunikace lze zajistit pomocí SSL, které běží nad samotným MQTT protokolem, kvůli zachování jednoduchosti a malého síťového provozu [3].



Obr. 9 - MQTT diagram [19]

6.3.2. Node-RED

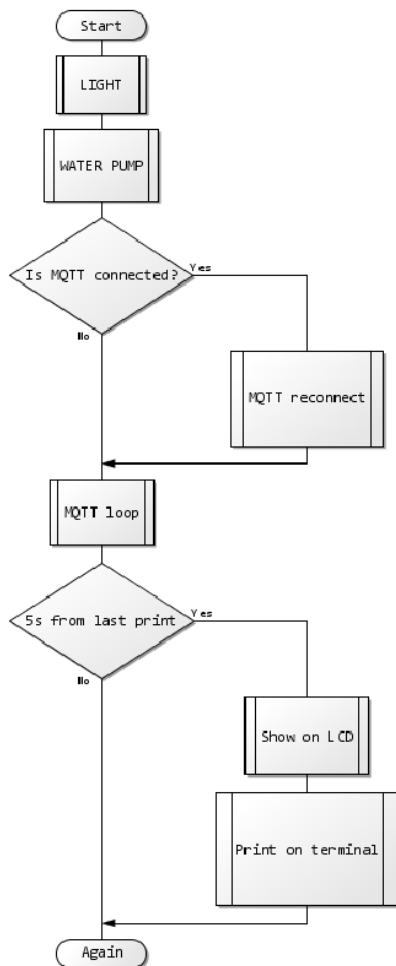
Node-RED je softwarový nástroj pro sjednocení a propojení všech hardwarových zařízení určených pro IoT, různých API a online služeb. Rozhraní přes webový prohlížeč poskytuje tvorbu jednotlivých procesů, které se dají vzájemně propojovat a kombinovat. Nasazení těchto procesů je pak otázkou jednoho kliknutí. Node-RED nabízí vlastní „konstrukční prvky“, které je možné okamžitě použít, širokou základnu knihoven, které se dají doinstalovat pomocí správy balíčků npm a dále funkční blok, kde si uživatel může pomocí JavaScriptu doprogramovat jakékoliv chování.

Celý Node-RED je postaven na Node.js, což s sebou přináší všechny výhody jeho event-driven a non-blocking modelu. Díky tomu je možné Node-RED nasadit na levný hardware, jako je například RaspberryPi.

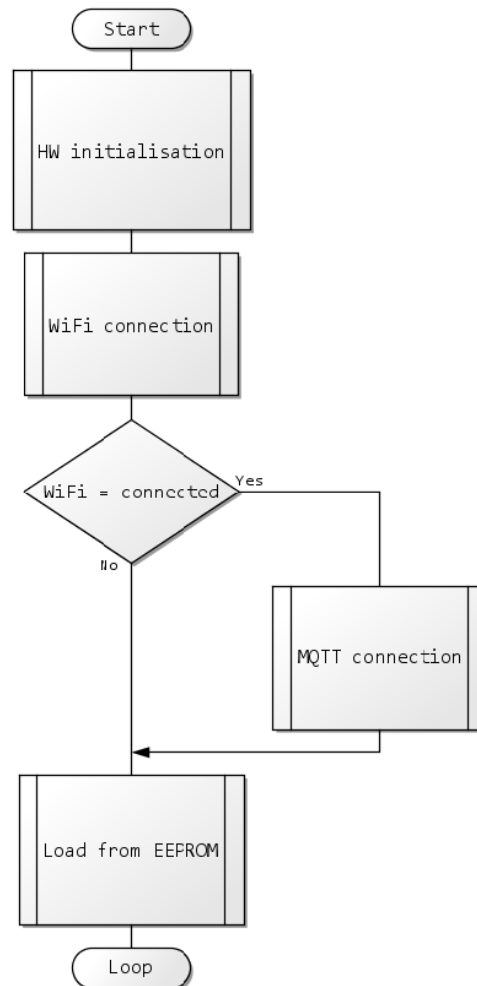
Veškeré procesy jsou po vytvoření a nasazení uloženy v JSON formátu, který je snadné importovat a exportovat pro účely zálohování, nebo sdílení s ostatními.

6.4. Arduino

Zdrojový kód pro Arduino se dá rozdělit do logických bloků propojených ve správném pořadí v hlavní metodě. Tyto bloky a logika celé aplikace jsou znázorněny v blokovém schématu. V prvním schématu je postup prvotního nastavení setup (Obr. 10) a v druhém schématu je pak opakující se cyklus běhu kódu nazývaný loop (Obr. 11).



Obr. 10 - setup



Obr. 11 - loop

6.4.1. Setup

Část programu nazývaná setup slouží k úvodnímu nastavení hardwaru. Je spuštěna pouze jednou a to hned po připojení napájení k mikrokontroleru.

6.4.2. HW initialisation

Zde je provedena inicializace komunikace se zařízeními (LCD, RTC, senzory) a nastavení IO pinů do příslušných režimů (vstup nebo výstup).

6.4.3. WiFi connection

Zde je proveden pokus o připojení k přednastavené síti WiFi a následně k MQTT serveru. Pokud připojení selže, je pokus ještě 10x opakován, poté systém přejde do režimu bez

spojení se serverem, kdy je načteno speciální nastavení pro samostatný běh a kód pokračuje do smyčky loop.

6.4.4. Load from EEPROM

Zde je provedeno načtení nastavení uloženého v EEPROM při předchozím spuštění do globálních proměnných.

6.4.5. Loop

Smyčka loop zajišťuje periodické volání funkcí nutných pro běh celého systému. Nejprve spustí funkce pro obsluhu svícení a zavlažování s parametry načtenými z EEPROM paměti.

Dále je provedena kontrola spojení s MQTT serverem, případně je provedeno i jeho opětovné připojení a následuje smyčka MQTT.loop, která zajišťuje příjem a obsluhu příchozích MQTT zpráv. Následně je provedena kontrola času od posledního výpisu dat na LCD displej a terminál a v případě, že je větší než 5 vteřin, data se aktualizují. Poté se opět celá tato smyčka zopakuje.

6.4.6. Funkce Light

Tato funkce se stará o správnou obsluhu osvitové jednotky. Funkce má tři režimy, které je možné libovolně měnit přes webové rozhraní v sekci nastavení.

První režim, který je defaultně nastaven je automatický režim, kdy jsou světla řízena na základě vypočteného času východu a západu slunce.

Druhý režim je manuální. Světla jsou jím řízena na základě času zapnutí a vypnutí v uživatelském nastavení.

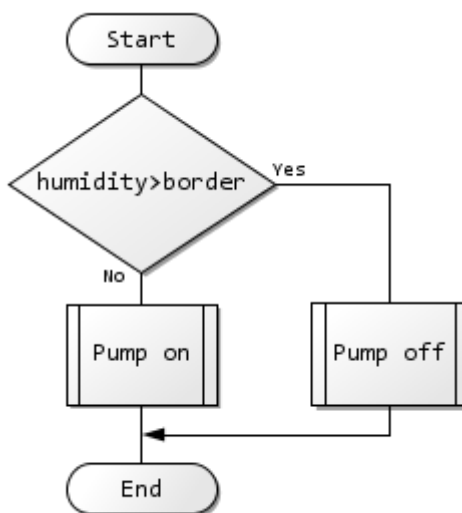
Třetí režim je tzv. override. V tomto režimu je uživateli umožněno přes webové rozhraní ručně světlo zapnout, či vypnout.

6.4.7. Funkce Pump

Toto je funkce, která má na starosti obsluhu čerpadla. Funkce má dva režimy, které je možné libovolně měnit přes webové rozhraní v sekci nastavení.

První režim, který je defaultně nastaven je automatický režim. V tomto režimu je nejprve provedena kontrola vlhkosti půdy a následně porovnána s nastaveným požadovaným prahem. V případě potřeby je pak spuštěna funkce zavlažování, která kromě zapínání a vypínání samotného čerpadla ještě kontroluje hladinu vody v nádrži (Obr. 12).

Druhý režim je opět override, který má stejnou funkčnost jako ve funkci Light.



Obr. 12 - Funkce Pump v automatickém režimu

6.4.8. MQTT

Pro potřeby použití MQTT protokolu na Arduinu byla využita knihovna Arduino PubSubClient, která se kompletně stará o připojení k MQTT serveru a následný příjem, nebo odesílání zpráv.

Podporuje obě právě používané verze MQTT protokolu (3.1 a 3.1.1) a autentifikaci pomocí jména a hesla. Bohužel není možné tuto knihovnu použít s šifrovaným režimem pomocí SSL/TLS, jelikož by na to nestačil výpočetní výkon Arduina [11].

6.4.9. MQTT reconnect

Tato funkce slouží k obsluze spojení MQTT klienta v Arduinu s MQTT serverem. Nejprve je proveden pokus o spojení s definovaným serverem a v případě, že se spojení povedlo navázat, je dále proveden subscribe jednotlivých MQTT topiců obsahujících nastavení pro zahradu a následně jsou na server odeslána debugovací data. V případě, že je z nějakého důvodu

nemožné se serverem spojení navázat, je tento pokus 10x opakován a následně systém přejde do samostatného režimu.

6.4.10. MQTT loop

Zde je obslužný kód, který zajišťuje příjem a obsluhu MQTT zpráv. Zprávy jsou nejprve rozříděny podle topicu a následně jsou převedeny do správného formátu a uloženy do korespondujících globálních proměnných. Poté je příchozí zpráva vytištěna na terminál a provede se uložení nastavení do paměti EEPROM a opětovné odeslání aktuálně používaného nastavení na server.

6.4.11. Show on LCD a Print on terminal

Zde je provedeno parsování dat právě používaného nastavení a dat ze senzorů a je ve správném formátu zobrazeno na displeji respektive vytištěno na terminál.

6.5. Raspberry Pi

6.5.1. Instalace operačního systému

Jako operační systém pro Raspberry Pi byla zvolena distribuce Linuxu [23] s názvem DietPi. Je to odlehčená verze Raspbianu Jessie, která plně vyhovuje potřebám této práce. Navíc obsahuje vlastní software pro instalaci různých často používaných balíčků a jejich automatickou konfiguraci, která je plně optimalizovaná pro použití s Raspberry Pi.

Samotná instalace spočívá ve stažení image systému z webových stránek projektu, následném zapsání na paměťovou kartu programem Win32 Disk Imager [20] a vložení do Raspberry Pi. Po připojení počítače do elektrické sítě je pak uživatel dotázán na několik základních parametrů nastavení a systém pak sám provede potřebnou konfiguraci a instalaci případných dalších SW komponent.

6.5.2. Změna hesla

Z bezpečnostních důvodů je dobré změnit defaultně používané heslo uživatele root, které je dietpi. K tomu slouží příkaz `passwd` následovaný dvojitým zadáním nového hesla.

6.5.3. SSH přístup

Pro možnost vzdáleného přístupu po síti byla využita technologie SSH, kterou je nutné nejprve doinstalovat a nastavit. Instalace byla provedena pomocí příkazu

```
dietpi-software
```

kde byl vybrán balíček OpenSSH k instalaci a následně opět pomocí

```
dietpi-software
```

byl vybrán jako SSH server.

6.5.4. Instalace LAMP

Instalace LAMP (Linux, Apache, MySQL, PHP) probíhala opět příkazem

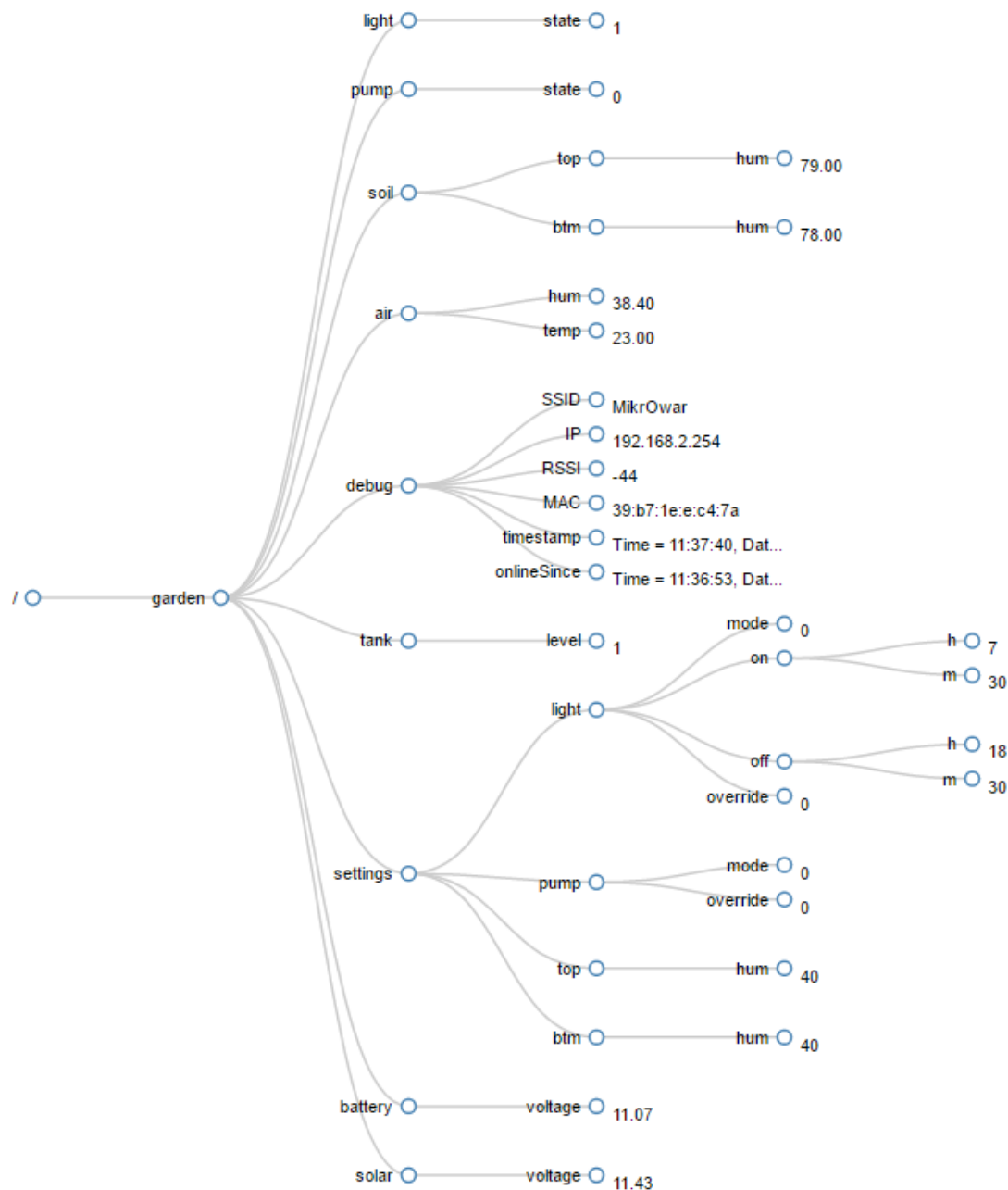
```
dietpi-software.
```

LAMP je v této práci použit pro účely TopicTree (viz 6.5.5. Mosquitto).

6.5.5. Mosquitto

Mosquitto je balíček umožňující spuštění MQTT serveru. Jeho instalace proběhla dle návodu přímo na stránkách výrobce [21]. Konfigurační soubor použitý pro potřeby této práce je v příloze (Příloha B).

Nyní po úspěšné instalaci a konfiguraci MQTT serveru je volitelným krokem použití Topictree, což je webová aplikace pro zobrazení jednotlivých topiců a jejich zpráv v hierarchii stromového grafu (Obr. 12). Topictree je ke stažení na GitHubu autora [22] a stačí ho pouze vložit do adresáře webového serveru na Raspberry Pi, v tomto případě do /var/www/ a pouze upravit v souboru index.html IP adresu MQTT serveru na IP právě přidělenou.



Obr. 12 - Topictree

6.5.6. Node-RED

Instalace balíčku Node-RED byla provedena přes balíčkovací nástroj apt příkazy

```
apt-get update
apt-get install nodered
```

Pro spuštění Node-RED po startu systému výrobce doporučuje použití příkazu,

```
systemctl enable nodered.service
```

který v tomto případě nefungoval a jako řešení byl tedy Node-RED přidán do rc.local pomocí textového editoru.

Node-RED umožňuje přidání různých funkčních prvků pomocí balíčkovacího nástroje npm, který byl doinstalován následujícími příkazy:

```
apt-get install npm
npm install -g npm@2.x
```

Pro potřeby uživatelského webového rozhraní plně vyhovuje balíček UI, který do Node-RED přidá funkční prvky grafů, ukazatelů, tlačítek, přepínačů a dalších. To vše s výhodami použití s Node-RED a šikovným responzivním designem. Jeho instalace přes npm probíhala takto:

```
cd /root
mkdir .node-red
cd ~/.node-red
npm install node-red-contrib-ui
```

Po samotné instalaci je nutné provést restart příkazy

```
node-red-stop
node-red-start
```

a balíček by měl být okamžitě dostupný.

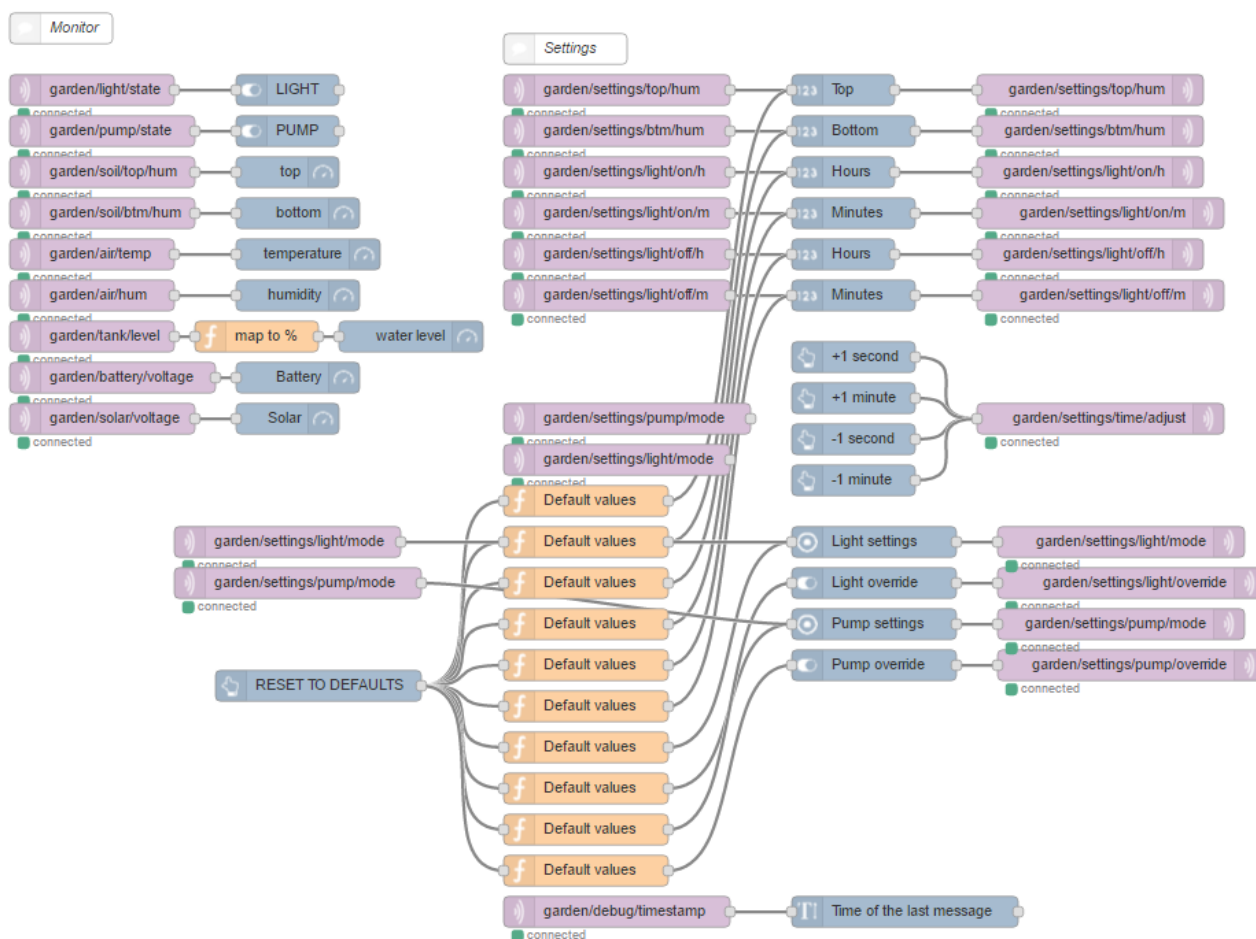
Node-RED má webové grafické rozhraní defaultně dostupné na adrese <http://localhost:1880/>, ve kterém je pak spojením jednotlivých bloků (Obr. 12) vytvořena logika celé aplikace (Obr. 13).



Obr. 13 - Stavbní bloky Node-RED

V této práci byly použity 3 typy stavebních bloků:

- Fialové představují MQTT komunikaci
- Oranžové jsou vlastní funkční bloky psané v jazyce JavaScript
- Modré jsou bloky prvků ve webovém rozhraní



Obr. 14 - Logika aplikace v Node-RED

7. Výsledky

7.1. Demonstrační prototyp

Demonstrační prototyp byl zhotoven z popsané elektroniky a jako stojan slouží na míru vyfrézovaný a sestavený stojan z laminované dřevotřísky (Obr. 15).



Obr. 15 - Hotový prototyp s detaily zavlažovací a osvitové jednotky



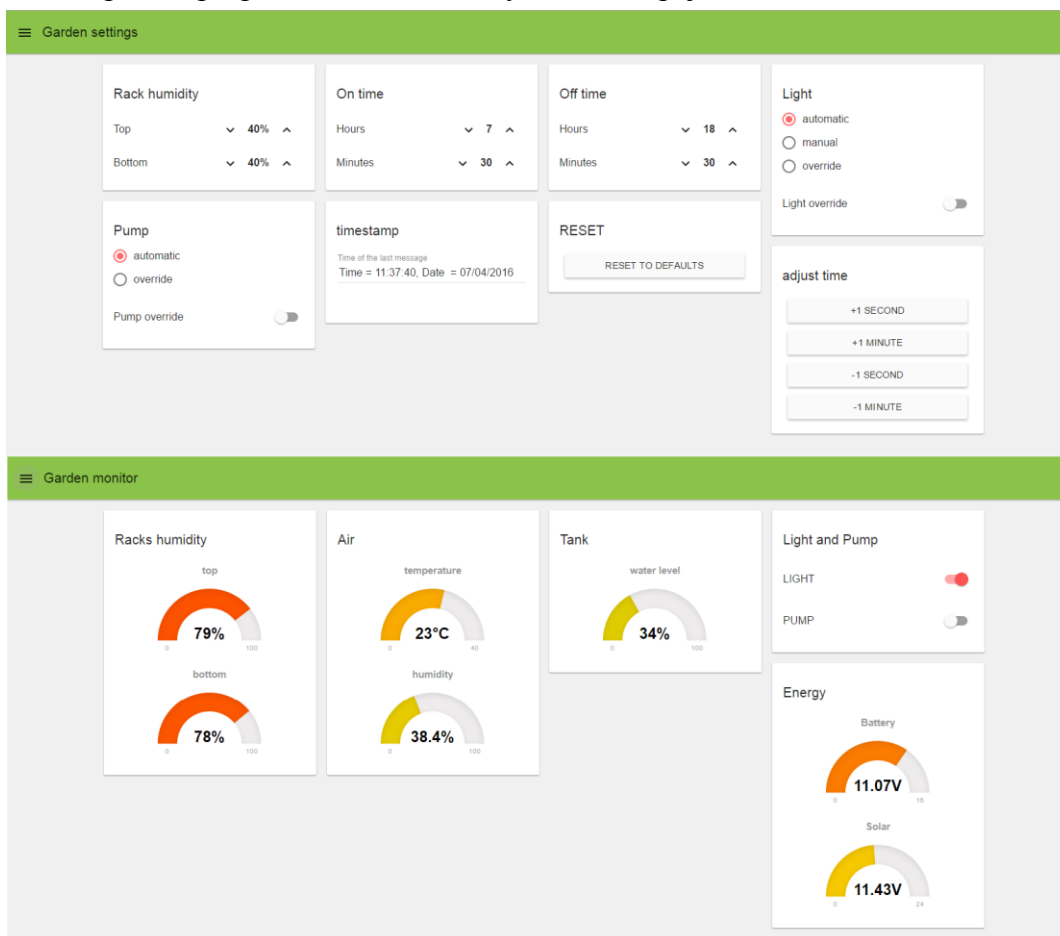
Obr. 16 - Detail technického zázemí

7.2. Webové rozhraní

Design webového rozhraní pro monitorovací a nastavovací aplikaci je rozdělen do jednotlivých bloků (Obr. 17).

V horní části je vidět menu uživatelského nastavení. V prvním bloku je nastavení požadované vlhkosti v květináči. Další dva bloky nastavují časy zapnutí a vypnutí přívitu v případě manuálního režimu, nebo si můžeme zvolit režim automatický, kdy se přívit zapíná a vypíná ve vypočteném čase východu a západu slunce. Jako poslední režim je zde režim override, kdy uživatel může manuálně přívit ovládat. Pak je zde nastavení režimu pro čerpadlo, buď automatické, nebo opět override. Dále je zde vidět čas poslední přijaté zprávy, tlačítko pro vrácení systému do defaultního nastavení a blok pro úpravu času v systému zahrady.

Ve spodní části obrázku je vidět monitorovací část aplikace. V prvním bloku jsou ukazatele vlhkosti půdy, v druhém bloku ukazatele teploty a vlhkosti vzduchu, další blok zobrazuje ukazatel výšky hladiny ve vodní nádrži a předposlední ukazatel zobrazuje stav světel a vodního čerpadla. Poslední blok na obrázku ukazuje napětí na akumulátoru a na solárním panelu při použití se solárním systémem napájení.



Obr. 17 - webové rozhraní

8. Testování

Při testování zhotoveného prototypu se vyskytly určité další problémy, které se dají rozdělit na dvě části:

- Problémy s hardwarem
- Problémy se softwarem

V následující části práce je jejich popis a možná řešení.

8.1. Řešené problémy s HW

Při použití některých HW prvků nižší kvality nastaly některé problémy, se kterými se při samotném návrhu systému nepočítalo, a tak je bylo nutné operativně vyřešit. Tyto problémy jsou níže popsány a jsou zde i nastíněna jejich možná řešení, případně řešení, která byla konkrétně použita.

8.1.1. Arduino webserver

Při návrhu první verze systému se nepočítalo s použitím dalšího HW pro webové rozhraní, ale předpokládalo se využití možnosti zřídit webový server přímo na Arduinu. To se částečně povedlo, ale výkon Arduina bohužel nebyl dostatečný, aby se tohoto úkolu zhostil s potřebnou rychlostí a stabilitou. Proto byl posléze realizován návrh a implementace, který je zde popsán.

8.1.2. WiFishield

Použití WiFi shieldu s sebou nese určité problémy, které se naštěstí dají vyřešit. Prvně měl být použit shield SparkFun WiFly Shield. Svými parametry by byl vyhovující, ale bohužel jeho vyvedené komunikační piny nejsou kompatibilní s topologií pinů na Arduinu Mega. Proto byl nakonec použit Arduino WiFi Shield, který je s Mega kompatibilní a splňuje veškeré požadavky.

Další věc, na kterou musí být brán ohled je obsazení některých pinů Arduina použitých pro komunikaci s tímto shieldem. V případě Arduino WiFi Shieldu to jsou konkrétně následující

piny: D10,D11,D12,D13, které jsou použité pro SPI komunikaci a dále piny D4 a D7 pro další jeho funkce.

8.1.3. Kalibrace senzorů napětí

Při použití odporového děliče jako senzoru napětí je nutné brát v potaz možnou toleranci hodnot použitých rezistorů. Proto je nutné provést kalibraci tohoto senzoru, aby nedocházelo k odchýlkám měření. Kalibrace se provádí změřením přesného odporu použitých rezistorů a úpravou hodnot ve zdrojovém kódu v části výpočtu vstupního napětí.

8.1.4. Kalibrace senzorů vlhkosti půdy

Senzory vlhkosti půdy, jak je napsáno výše, pouze měří vodivost mezi dvěma elektrodami. Pro výpočet přesné vlhkosti půdy je tedy nutné porovnat hodnoty naměřené tímto senzorem a naměřené profesionálním měřičem vlhkosti půdy.

8.1.5. Opotřebení kontaktů senzorů vlhkosti

Při vložení senzorů vlhkosti do vlhké půdy zde bude docházet k jejich korozi. Koroze bude navíc urychlena procházejícím elektrickým proudem mezi elektrodami. Proto zde byla použita dvě protiopatření. Pro snížení koroze vlivem elektrického proudu je senzor napájen pouze při samotném čtení, které probíhá v delších intervalech. Navíc mohou být elektrody opatřeny snímatelnými kovovými hroty, pro možnost výměny v případě velkého opotřebení.

8.1.6. Nedostatečný výkon solárního panelu

Při experimentu se solárním napájením byl použit dostupný solární panel o výkonu 40 W. Tento výkon není bohužel plně dostačující pro napájení celého systému řízení komunitní zahrady, zvláště pak osvitové jednotky. Systém, tak jak je navržen, je připraven pro použití i s výkonnějším fotovoltaickým panelem, nebo s kombinací více kusů panelů, aby bylo možné plně dobít použitý akumulátor, ze kterého by pak byl systém napájen. Pro potřeby demonstračního přípravku je však zbytečné a finančně příliš nákladné přidávat další panely, a tak je osvitová jednotka samostatně napájena z elektrické sítě.

8.2. Řešené problémy v Arduinu

Při psaní zdrojového kódu pro mikrokontroler mohou nastat problémy, které je nutné průběžně řešit. Pro mnoho funkčních problémů je již dnes možné najít hotové řešení na internetu. Následující problémy však stojí za zmínku a jsou zde proto podrobněji popsány i možnosti jejich řešení.

8.2.1. Přerušené spojení

Při chybách na síti mezi serverem a zahradou by mohlo dojít k nepříjemným problémům. Například situace, kdy uživatel manuálně zapne čerpadlo zavlažovacího systému a v tu chvíli se přeruší spojení, by mohla vést k nenávratnému poškození celé zahrady, případně i dalšího majetku v okolí.

Řešením tohoto problému je včasná detekce přerušení spojení a následné přepnutí kritických částí do plně automatických režimů.

8.2.2. Synchronizace času pomocí NTP

Pro synchronizaci času zahrady by bylo vhodné použít NTP synchronizaci. Bohužel se při testování demonstračního přípravku v chodu ukázalo, že při použití NTP synchronizace se Arduino stalo nestabilním a občas se zaseklo.

Proto bylo nakonec použito řešení, kdy uživatel sám skrz webové nastavení je schopen upravit přesný čas v zahradě.

8.2.3. Kontrola času v daném intervalu

Pro možnost automatického a poloautomatického režimu svícení je nutné přesně stanovit, jestli se aktuální čas nachází v nastaveném intervalu, nebo mimo něj. Pro tento požadavek byla napsána funkce, která právě toto kontroluje pomocí následující části zdrojového kódu:

```
bool timeIsBetween(int leftIntervalHour, int leftIntervalMinute, int rightIntervalHour, int rightIntervalMinute) {  
    int from = leftIntervalHour * 100 + leftIntervalMinute;  
    int to = rightIntervalHour * 100 + rightIntervalMinute;  
    int actual = tm.Hour * 100 + tm.Minute;  
    return to > from & actual >= from & actual <= to | to < from & (actual >= from | actual <= to);  
}
```

8.2.4. Šifrování MQTT spojení

Z bezpečnostních důvodů by bylo vhodné šifrovat komunikační kanál mezi serverem a zahradou. Případný útočník by nejen mohl odchyťvat, ale hlavně i měnit zasílaná nastavení a způsobit tak potenciální škody. Arduino však nemá dostatečný výkon pro jakékoliv šifrování a tak nebylo v demonstračním přípravku použito.

9. Závěr

V úvodu této práce byly uvedeny tři hlavní cíle:

- Analýza hotových systémů a produktů pro řízení komunitních zahrad
- Návrh vlastního systému
- Realizace demonstračního přípravku

Všechny tyto cíle byly splněny. Po provedené analýze v praktické části jsou mimo jiné uvedeny nastalé problémy v průběhu realizace demonstračního přípravku a jejich způsoby řešení. Zařízení bylo v praxi odzkoušeno a při jeho běhu se neprojevil žádný další problém.

Tento projekt kromě použití pro řízení komunitní zahrady má i potenciál pro řízení klimatu v teráriích, paludáriích a akváriích po příslušném upravení vstupních a výstupních komponent.

Jako další možnosti rozšíření by bylo možné v případě použití s Raspberry Pi v místě zahrady instalovat webkameru pro online přenos růstu rostlin, nebo například i systém ultrazvukového měření jejich výšky.

10. Seznam literatury

1. SRBENÝ, Lukáš. Management tvorby komunitních zahrad - sídliště Máj České Budějovice. Praha, 2014.
2. Node-RED [online]. [cit. 2016-02-28]. Dostupné z: <http://nodered.org/>
3. MQTT [online]. [cit. 2016-02-28]. Dostupné z: <http://mqtt.org/>
4. BANZI, Massimo a Michale SHILOH. GettingStartedwithArduino. 3rd Edition. San Francisco, CA: Maker Media, Inc, 2014. ISBN 978-1-4493-6333-8.
5. *GitHub* [online]. [cit. 2016-02-28]. Dostupné z: <https://github.com/knolleary/pubsubclient>
6. *Arduino* [online]. [cit. 2016-03-12]. Dostupné z: <https://www.arduino.cc/en/Main/ArduinoBoardUno>
7. *AMC2302 datasheet* [online]. [cit. 2016-03-12]. Dostupné z: <https://www.adafruit.com/datasheets/Digital%20humidity%20and%20temperature%20sensor%20AM2302.pdf>
8. *Moisture sensor datasheet* [online]. [cit. 2016-03-12]. Dostupné z: ftp://imall.iteadstudio.com/Electronic_Brick/IM121017001/DS_IM121017001.pdf
9. *Grovelabs* [online]. [cit. 2016-04-04]. Dostupné z: <https://grovelabs.io/>
10. *Clickandgrow* [online]. [cit. 2016-04-04]. Dostupné z: <http://www.clickandgrow.com/>
11. *ArduinoPubSubClient* [online]. [cit. 2016-04-04]. Dostupné z: <http://www.hivemq.com/blog/mqtt-client-library-encyclopedia-arduino-pubsubclient/>
12. *WiFiShield*[online]. [cit. 2016-04-20]. Dostupné z:
[z:https://www.arduino.cc/en/Main/ArduinoWiFiShield](https://www.arduino.cc/en/Main/ArduinoWiFiShield)
13. *AMC2302* [online]. [cit. 2016-04-20]. Dostupné z: http://www.getmicros.net/wp-content/uploads/2013/12/Arduino-and-DHT11_bb.png
14. *CLICK and GROW* [online]. [cit. 2016-04-20]. Dostupné z: http://cdn.shopify.com/s/files/1/0156/0137/products/NewProject1_1_1024x1024.gif?v=1456481310
15. *Grove* [online]. [cit. 2016-04-20]. Dostupné z: <https://things.grovelabs.io/craft/home/ecosystem-product.png>
16. *LCD I²C* [online]. [cit. 2016-04-20]. Dostupné z: http://www.4tronix.co.uk/arduino/pictures/i2c_03.jpg

17. *RTC* [online]. [cit. 2016-04-20]. Dostupné z:
http://www.tuelectronica.es/images/tutoriales/arduino/rtc_ds1307/RTC_DS1307_08.gif
18. Solar regulator datasheet [online]. [cit. 2016-04-20]. Dostupné z:
http://www.aspa.cz/doc/d/o/c/carspa_pdf.pdf
19. *MQTT diagram* [online]. [cit. 2016-04-20]. Dostupné z: <http://www.4makers.info/wp-content/uploads/2016/03/komarSchema.jpg>
20. *Win32 Disk Imager* [online]. [cit. 2016-04-20]. Dostupné z:
<https://sourceforge.net/projects/win32diskimager/>
21. *Mosquitto* [online]. [cit. 2016-04-20]. Dostupné z:
<http://mosquitto.org/2013/01/mosquitto-debian-repository/>
22. *Topic tree* [online]. [cit. 2016-04-20]. Dostupné z: <https://github.com/hardillb/d3-MQTT-Topic-Tree>
23. *DietPi* [online]. [cit. 2016-04-20]. Dostupné z: <http://dietpi.com/>

11. Seznam příloh

Příloha A - Slovník zkratk a pojmů

Příloha B - Obsah přiloženého CD

- text diplomové práce ve formátu pdf
- zdrojový kód pro Arduino
- zdrojový kód z Node-RED ve formátu JSON
- Konfigurační soubor mosquito.conf

Příloha A

Slovník zkratk a pojmů:

- MQTT–Message Queuing Telemetry Transport
- LED–Light Emitting Diode
- IoT– Internet of Things
- npm– Balíčkovací nástroj JavaScriptových aplikací
- JSON–Java Script Object Notation
- UART–universal asynchronous receiver/transmitter
- I²C–Inter-IntegratedCircuit
- SPI–Serial Peripheral Interface
- driver– pomocný řídicí obvod
- HW– hardware
- SW– software
- CRI–color rendering index
- PWM–Pulse-width modulation
- M2M–Machine to machine
- IO–Input Output
- EEPROM–Electrically Erasable Programmable Read-Only Memory
- SRAM–Static Random Access Memory
- A/D– Analog digital