



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATIZOVANÉ PĚSTOVÁNÍ ROSTLIN

AUTOMATED PLANT GROWING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAN ŠVANCER

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MARTIN MUSIL

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

Zadání diplomové práce

Řešitel: **Švancer Jan, Bc.**

Obor: Počítačová grafika a multimédia

Téma: **Automatizované pěstování rostlin
Automated Plant Growing**

Kategorie: Vestavěné systémy

Pokyny:

1. Seznamte se s problematikou Internet-of-Things se zaměřením na automatické řízení závlahových systémů, pěstebních stanů/boxů, hydroponických systémů a dalších způsobů automatizované podpory růstu rostlin.
2. Navrhněte konkrétní úlohu, kterou budete v rámci práce realizovat. Shrňte možnosti realizace bezdrátové sítě senzorů a řídicích modulů (např. pomocí XBee, Bluetooth, Wi-fi a dalších). Navrhněte jejich hardwarovou realizaci na základě existujících hardwarových modulů a platform.
3. Navržené řešení realizujte i po hardwarové stránce. Implementujte pokročilejší funkce pro automatizované řízení závlivky na základě vnějších veličin a údajů z bezdrátových senzorů.
4. Realizujte připojení celého systému do Cloudu a umožněte vzdálený přístup a řízení pomocí mobilní aplikace.

Literatura:

- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Musil Martin, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 24. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato práce se zabývá spojením dvou současných trendů – internetu věcí a domácího pěstování zeleniny, ovoce a bylinek. Popisuje technologické možnosti pro bezdrátovou komunikaci senzorů v chytré domácnosti a hardwarové prostředky pro realizaci. Na jejich základě realizuje chytrý hardware pro automatizované pěstování domácích rostlin, připojený do cloudu se vzdáleným přístupem pomocí iOS mobilní aplikace.

Abstract

This thesis combines two current trends – the Internet of Things and Urban Agriculture. It describes technologies for wireless sensors communication in smart homes and hardware for their implementation. Based on these technologies smart hardware for automatic plants growing is implemented and connected to cloud with remote access via iOS mobile application.

Klíčová slova

Internet věcí, chytrá domácnost, domácí automatizace, hydroponické a závlahové systémy, Arduino, ATmega328, Wi-Fi, ESP8266, NodeMCU, NRF24L01, iOS, mobilní aplikace, Swift, Firebase

Keywords

Internet of Things, smart home, home automation, hydroponic and irrigation systems, Arduino, ATmega328, Wi-Fi, ESP8266, NodeMCU, NRF24L01, iOS, mobile application, Swift, Firebase

Citace

ŠVANCER, Jan. *Automatizované pěstování rostlin*. Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Musil Martin.

Automatizované pěstování rostlin

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Martina Musila. Další informace mi poskytl pan Ing. Petr Musil. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Švancer
19. května 2017

Poděkování

Chtěl bych poděkovat vedoucímu práce panu Ing. Martinu Musilovi, díky kterému mohla práce vzniknout. Poskytl mi nejen odbornou pomoc, ale také pomoc s fyzickým osazováním vyrobených desek plošných spojů. Dále bych rád poděkoval firmě Speco Control s.r.o., která mi umožnila v jejich dílnách realizovat část hardwarového řešení. Poděkování patří i mému kamarádovi Martinu Taťákovi, za pomoc s navržením desky plošných spojů a umožnění přístupu k měřicím přístrojům společnosti Honeywell.

Obsah

1	Úvod	5
2	Internet věcí a domácí automatizace	6
2.1	Úvod k internetu věcí	6
2.2	Cloudové služby v IoT	7
2.3	iOS mobilní aplikace	9
2.4	Hydroponické a závlahové systémy	10
2.5	Přehled zařízení na trhu	10
3	Bezdrátová komunikace a dostupný hardware pro IoT	15
3.1	Komunikační protokoly	15
3.2	Standardy pro bezdrátovou komunikaci	18
3.3	Komunikační moduly	23
3.4	Jednodeskové počítače	25
3.5	Senzory	28
3.6	Akční členy	30
4	Specifikace systému pro automatizované pěstování rostlin	31
4.1	Bezdrátová síť	31
4.2	Gateway modul (koordinátor sítě)	33
4.3	Půdní senzor	34
4.4	Akční modul pro automatizované pěstování rostlin	34
4.5	Cloud a mobilní iOS aplikace	35
5	Návrh a realizace systému	36
5.1	Bezdrátová lokální síť	36
5.2	Datový model	40
5.3	Gateway modul	41
5.4	Půdní senzor	44
5.5	Akční modul	50
5.6	Firebase a logika systému	53
5.7	iOS mobilní aplikace	56
6	Testování a finanční zhodnocení	61
6.1	Test bezdrátové komunikace	61
6.2	Testování modulů a celého systému	62
6.3	Finanční zhodnocení	65

7 Závěr	67
Literatura	69
Přílohy	71
A Obsah přiloženého paměťového média	72
B ER diagram datového modelu	73
C Diagram případů užití mobilní aplikace	74
D Elektrická schémata	75
D.1 Gateway modul	75
D.2 Půdní senzor	76
D.3 Akční modul	77
E Podklady pro výrobu desky plošných spojů	78

Seznam obrázků

2.1	Oblasti internetu věcí (zdroj: http://saphanatutorial.com)	7
2.2	Amazon AWS IoT (zdroj: Amazon)	8
2.3	Rachio 2nd Generation (zdroj: Amazon)	11
2.4	Niwa One (zdroj: Niwa)	12
2.5	vlevo Leaf (zdroj: Leaf), vpravo Click and Grow – Smart Garden (zdroj: Click and Grow)	13
2.6	vlevo Parrot Flower Power (zdroj: Parrot), vpravo Xiaomi Mi Plant (zdroj: Xiaoland)	14
3.1	Formát zprávy CoAP (zdroj: IETF)	16
3.2	Hierarchická struktura témat v MQTT (zdroj: HiveMQ.com)	17
3.3	MQTT publish – subscribe model (zdroj: HiveMQ.com)	18
3.4	Bluetooth Stack (zdroj: Bluetooth.com)	19
3.5	Architektura Wi-Fi sítě (zdroj: Eprin.cz)	21
3.6	OSI model protokolu ZigBee (zdroj: vyvoj.hw.cz)	22
3.7	vlevo HC-05 modul, vpravo esp8266 (zdroj: santy.cz)	23
3.8	XBee moduly s různými typy antén (zdroj: digi.com)	25
3.9	Raspberry Pi 3 Model B (zdroj: Raspberrypi.org)	26
3.10	BeagleBone Black (zdroj: Beagleboard.org)	27
3.11	zleva Arduino Uno, Arduino Mega a Arduino Nano (zdroj: Arduino.cc)	28
3.12	zleva senzor půdní vlhkosti, DHT-22 a HC-SR04 (zdroj: Ebay.com)	29
3.13	Relé modul připojený k Arduino Uno (zdroj: Browse-tutorials.com)	30
4.1	Uvažované topologie bezdrátové sítě	32
4.2	Modul NodeMCU (zdroj: arduinotech.cz)	33
4.3	Nákres systému pro automatizované pěstování rostlin	35
5.1	Zapojení komunikačního modulu NRF24L01 s Arduino Uno	37
5.2	Párování a zrušení párování uzlů v síti	39
5.3	Gateway modul	43
5.4	Prototyp půdního senzoru	45
5.5	Osazená deska plošných spojů půdního senzoru	46
5.6	3D model	47
5.7	Půdní senzor	48
5.8	Vybíjecí křivka baterie	49
5.9	Zásobník vody	50
5.10	Řídící box akčního modulu	51
5.11	Vnitřní zapojení řídicího boxu	52
5.12	Ukázka JSON struktury databáze	54

5.13	Hlavní sekce aplikace	57
5.14	Detail senzoru, přidání (spárování) nové rostliny a detail druhu rostliny . .	58
5.15	Ukázka uživatelského rozhraní v nástroji Storyboard	59
6.1	Průměrný odběr elektrického proudu půdního senzoru	63
6.2	Testování automatického systému	64
6.3	Grafy naměřených hodnot z půdního senzoru	65

Kapitola 1

Úvod

Pojem Internetu věcí se v poslední době dostává stále více a více do povědomí široké veřejnosti. Ačkoliv se jedná o velmi diskutované téma, jeho přesná definice zatím neexistuje. Zjednodušeně tím lze rozumět miliony věcí připojených do globální sítě internetu za účelem sběru, výměny a vyhodnocení dat. Danými věcmi může být v podstatě cokoliv od miniaturních senzorů, přes chytré elektrospotřebiče až po obří letadla. Tím se otevírá využití internetu věcí v mnoha oblastech lidské činnosti. Jednou, pro obyčejného člověka asi nejvýraznější aplikací, jsou chytré domácnosti. Dalším trendem poslední doby je zdravý životní styl a s tím spojené správné stravování. Lidé se více zajímají o to, co kupují a také si začínají některé potraviny pěstovat sami doma, ve městech. Tento trend se nazývá anglickým názvem *Urban Agriculture*. Tato práce se zabývá spojením těchto dvou trendů, tedy domácího pěstování zeleniny, ovoce či bylinek za pomoci moderních technologií. Je zde popsán kompletní návrh automatizovaného systému pro pěstování. Od počáteční inspirace, přes návrh, až po jeho fyzickou realizaci, včetně uživatelského rozhraní v podobě iOS mobilní aplikace.

Na začátek v kapitole 2 je představen koncept internetu věcí spolu s využitím cloudových služeb a mobilních zařízení jako uživatelského rozhraní. Dále jsou zde popsány výhody hydroponických a závlahových systémů a představena vybraná řešení, která se v současné době nacházejí na trhu, nebo byly alespoň představeny veřejnosti, ale dosud neuvedeny do prodeje. Další kapitola 3 se zabývá technologiemi pro bezdrátovou komunikaci, která je nejčastěji použita pro komunikaci senzorů a akčních prvků v chytré domácnosti. Také jsou zde popsány různé hardwarové komponenty související s danou problematikou, jako komunikační moduly, jednodeskové počítače, platforma Arduino a další. Na základě těchto informací je v kapitole 4 představen návrh konkrétního systému pro realizaci, včetně výběru základních technologií, na kterých bude systém postaven. Detailní návrh a realizace jsou popsány v kapitole 5, kde jsou rozebrány jednotlivé komponenty systému, jejich hardwarová i softwarová realizace, použité nástroje a logika systému. Realizovaný systém byl následně otestován z několika hledisek. Postupy a získané výsledky jsou shrnuty v kapitole 6, kde se také nachází krátké ekonomické zhodnocení. V závěru práce 7 je celý projekt zhodnocen a nastíněn možný budoucí vývoj.

Kapitola 2

Internet věcí a domácí automatizace

Tato kapitola se zabývá úvodem do internetu věcí ve spojení s využitím cloudových služeb. Dále se zaměřuje na problematiku hydroponických a závlahových systémů a to jak do interiéru, tak exteriéru. V závěru jsou představeny vybraná zařízení, která se v této oblasti již vyskytují.

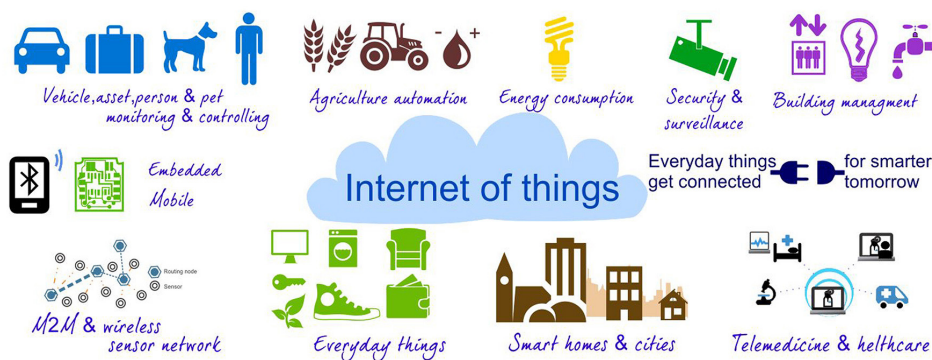
2.1 Úvod k internetu věcí

Internet věcí (anglicky Internet of Things, zkráceně **IoT**) je relativně nový trend v oblasti elektrotechniky s obrovským potenciálem do budoucna. Zjednodušeně řečeno se jedná o propojení fyzických zařízení s internetem za účelem sběru, výměny a vyhodnocení dat. Fyzickými zařízeními rozumíme v podstatě cokoli – jednoduché senzory, domácí elektrospotřebiče (ledničky, pračky), automobily, nebo třeba dopravní semaforey. Zařízení zpravidla obsahují senzory pro monitorování stavů a veličin, akční prvky pro vykonávání událostí a komunikační rozhraní. [13, 22]

Důležitou technologií pro vznik internetu věcí se považuje vynález RFID¹. RFID je technologie identifikátorů sloužící například k identifikování a sledování poštovních zásilek, sledování dobytka, k ochraně zboží proti krádeži v obchodech, nebo třeba k placení mýta na dálnicích. Kořeny technologie sahají do doby druhé světové války. Ovšem až v roce 1973 byl udělen první americký patent na aktivní RFID tag s přepisovatelnou pamětí. Zásadním krokem bylo založení *Auto-ID Center* a provázání RFID s internetem v roce 1999. *Auto-ID Center* vytvořil globální standard pro RFID a další senzory. Spolu s touto událostí se poprvé více objevuje termín Internet věcí použitý jedním ze zakladatelů *Auto-ID Center* Kevinem Ashtonem. [14]

Internet věcí přináší nové možnosti v nejrůznějších oblastech jako doprava a logistika, zdravotnictví, nebo energetika. Dále se tato technologie uplatňuje v inteligentních domácnostech. Příkladem zde mohou být inteligentní termostaty, osvětlení či chytré zásuvky. Do budoucna se předpokládají i takové vize jako chytré ledničky, které sami dokáží zjistit svůj obsah a na základě naučeného chování uživatele objednat chybějící potraviny. Jiné senzory mohou kontrolovat klima v lednici a identifikovat zkažené potraviny. Přehled oblastí, kde nachází internet věcí své uplatnění, je ilustrován na obrázku 2.1. [12]

¹Radio Frequency Identification



Obrázek 2.1: Oblasti internetu věcí (zdroj: <http://saphanatutorial.com>)

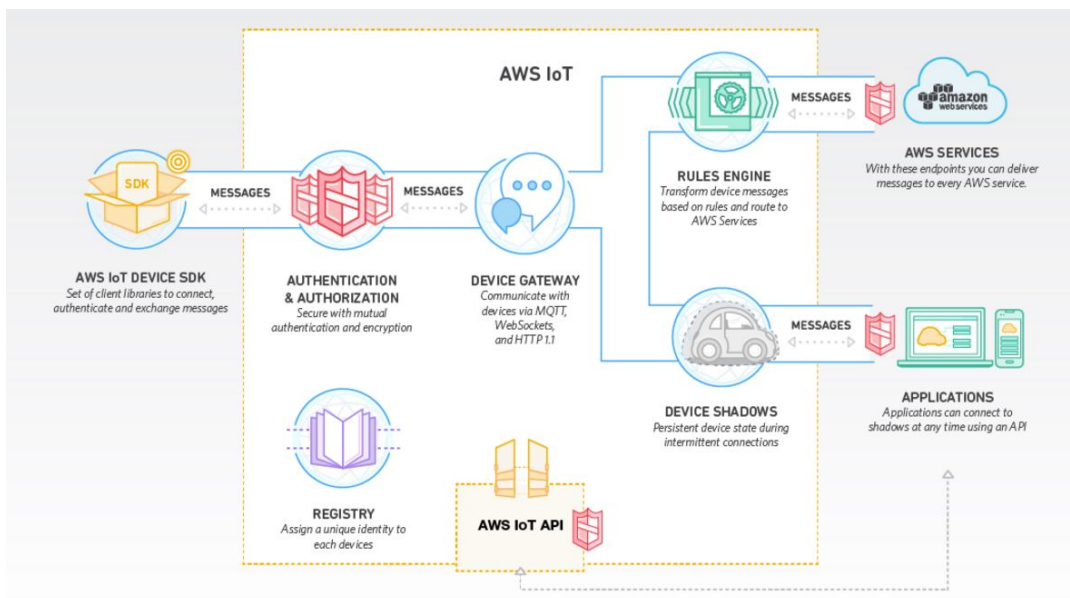
Ačkoliv se jedná o v poslední době velmi diskutované téma, stále není specifikován jednotný protokol a standard. Vyskytuje se tak řada odlišných protokolů, které mohou do celé situace vnést jistou nekonzistenci. Zařízení od jednoho výrobce pak nemusí komunikovat s jinými zařízeními a tím se lehce vytrácí myšlenka globální propojené sítě internetu věcí.

2.2 Cloudové služby v IoT

Samotné připojení reálných věcí k internetu ještě ovšem nevytváří nic speciálního. Nedílnou součástí celého konceptu internetu věcí je sběr a vyhodnocování nasbíraných dat, jakožto sdílení těchto dat s dalšími službami. Například řada dnešních automobilů obsahuje senzory a elektroniku, která dokáže sama identifikovat závadu. Pokud by tyto technologie byly připojené k internetu a propojené s autoservisy a dodavateli náhradních dílů, bylo by možné automaticky identifikovat závadu na automobilu, objednat vadný díl a doručit jej do autoservisu ještě dříve, než zákazník vůbec přijede. Jiný příklad využití může být v případě autonehody. Vozidlo a senzory v nositelné elektronice dokáží sami zjistit rozsah škody, zdravotní stav řidiče a na základě těchto informací přivolat pomoc, odeslat informaci o pacientovi do nemocnice a nahlásit pojistnou událost na pojišťovně.

Aby bylo výše uvedené možné, využívá se technologie Cloud Computingu. Tedy ukládání a zpracování dat na internetu, namísto lokálního úložiště, sdílení informací a nabízení služeb, či celých systémů běžících v cloudu. Pro internet lidí existuje v současné době již řada Cloudových služeb, jako *Google Drive*, *iCloud* nebo *Amazon Cloud Drive*. Tyto služby poskytují většinou kromě samotného úložiště pro data i celé aplikace pro běžné uživatele. Příkladem jsou Google Docs, Google Sheets, nebo Office Online. Pro internet věcí takové služby teprve pomalu vznikají a dostávají se do popředí. Mezi první poskytovatele Cloudových služeb vytvořených speciálně pro internet věcí patří typicky velké firmy jako Amazon, Microsoft, Google, nebo Oracle. Ukázkou struktury Cloudové služby Amazon AWS IoT lze vidět na obrázku 2.2. [10, 11]

Kritickým prvkem připojení zařízení do Cloudu je bezpečnost. Všechny zařízení musí být na obou stranách komunikace autorizovány a autentifikovány a data přenášena v šifrované podobě. Další součástí internetu věcí je samotné zpracování a vyhodnocení získaných dat. Jelikož se mnohdy jedná o obrovské množství informací, můžeme zde hovořit o pojmu *Big data*. Systém na základě vstupních hodnot produkuje události, kterými může být sepnutí/vypnutí akčních prvků, odeslání informační zprávy a jiné. Koncept internetu věcí



Obrázek 2.2: Amazon AWS IoT (zdroj: Amazon)

předpokládá autonomní rozhodování bez zásahu člověka. Ten pouze definuje pravidla pro chování a dále se již o systém nestará. Aby ovšem bylo možné systém monitorovat a případně pravidla měnit, využívá se mimo jiné v dnešní době již velmi rozšířených mobilních zařízení. Ty nám poskytují vzdálenou správu celého systému z libovolného místa na Zemi. [10]

ThingSpeak

ThingSpeak je cloudová služba od společnosti MathWorks, Inc. určená speciálně pro internet věcí. Dovoluje sběr velkého množství dat ze senzorů do cloudu, jejich vizualizaci a hlavně pokročilé analytické funkce, díky propojení se službou MATLAB. Data jsou ukládána do soukromých nebo veřejných kanálů a přístupná skrze webové rozhraní nebo API. Služba také dovoluje vykonávat akce na základě vyhodnocených dat a je v základní verzi zdarma s omezením.

Google Firebase

Další zajímavou službou je Firebase od společnosti Google. Tato služba není primárně určena pro internet věcí a práci s velkým množstvím dat, jako například ThingSpeak, ale poskytuje jednoduchý backend pro mobilní a webové aplikace. V oblasti internetu věcí se používá nejčastěji v kombinaci s jinými službami od společnosti Google, kdy naměřená data jsou ukládána do služeb Google Storage (Cloud Storage, Cloud Datastore, Cloud BigTable) pomocí Cloud Pub/Sub, zpracovávána a analyzována pomocí Google Analytics (Cloud Dataflow, BigQuery, Cloud Dataproc, Cloud Datalab) a pouze aktuální stav je zrcadlen ve službě Firebase.

Firebase poskytuje řadu několika služeb, mezi které patří:

- **Firestore Analytics** – analytické nástroje podobné Google Analytics. Obsahuje měření o uživateli, používání aplikace či nákupech a marketingových kampaních.

- **Authentication** – služba pro autentifikaci uživatelů. Kromě přihlášení pomocí emailu a hesla, podporuje také celou řadu sociálních sítí jako Facebook, Google, Twitter, GitHub a další.
- **Realtime Database** – databáze v reálném čase. Nejedná se o relační databázi, ale data jsou ukládána a adresována v JSON struktuře. Podporovány jsou tedy pouze základní datové typy jako čísla a textové řetězce. Ke službě existuje nativní SDK pro iOS, Android a JavaScript, které udržuje databázi v aplikaci vždy aktuální. Není tak potřeba manuální synchronizace dat a to dokonce v případě, kdy aplikace není aktuálně připojena k internetu. Kromě SDK existuje také REST API pro přístup k databázi. Dále lze definovat pravidla pro zajištění správnosti datových typů, přístup uživatelů a jiné.
- **Cloud Storage** – datové úložiště pro soubory. Opět poskytuje SDK pro iOS, Android a JavaScript.
- **Cloud Functions** – služba umožňuje definovat akce v jazyce JavaScript na základě vzniklých událostí. Událostí může být zápis dat do databáze, uložení souboru do úložiště, přihlášení uživatele, HTTP událost a další. Díky této službě odpadá potřeba vlastního serveru pro zajištění logiky aplikace. Jedná se o novinku, která byla představena teprve začátkem roku 2017.
- **Cloud Messaging** – multiplatformní řešení pro zasílání zpráv a notifikací mezi zařízeními. Umožňuje zasílání zpráv konkrétním uživatelům, či celým skupinám například na základě výsledků ze služby Analytics.

2.3 iOS mobilní aplikace

iOS je mobilní operační systém od společnosti Apple Inc. První verze tohoto systému byla představena v roce 2007, spolu s prvním telefonem iPhone. Systém byl původně určen pouze pro mobilní telefony iPhone a multimediální přehrávače iPod Touch, postupem času se však rozšířil i na další Apple zařízení, především tablet iPad. Operační systém iOS je systém UNIXového typu. Jedná se o odlehčenou verzi operačního systému MAC OS, používaného v počítačích Apple. Jelikož je určen pro mobilní zařízení, neobsahuje veškerou funkcionalitu MAC OS, na druhou stranu ale přidává podporu dotykového ovládání.

Vývoj mobilních aplikací pro iOS probíhá v nativním prostředí **Xcode** a je možný v jazyce **Objective-C** nebo **Swift**, který byl představen na konferenci WWDC v červnu 2014. Oba jazyky jsou objektově orientované, přičemž Swift je zamýšlen jako moderní náhrada za starý Objective-C. Swift je relativně mladý a jeho syntaxe se stále vyvíjí. Aktuální verze Swift 3 obsahuje obrovské množství změn oproti předchozím verzím a celkově konceptu programování iOS aplikací. Například zažitým stylem pojmenování funkcí byly dlouhé, vše vysvětlující, názvy jako `text.stringByTrimmingCharactersInSet`, Swift přináší změnu na krátké názvy (`text.trimming`), doplněné o parametry. Syntaxe jazyka je popsána v oficiální publikaci *The Swift Programming Language* [5]. Základními frameworky pro iOS aplikace jsou Cocoa a Cocoa Touch, který poskytuje nástroje pro práci s dotykovým rozhraním mobilního zařízení. Při programování se využívá koncepce **MVC** – Model-View-Controller.

K vývoji aplikací pro Apple (iOS, MAC OS) je nutné vlastnit vývojářskou licenci. Licence je placená a platí se vždy na 1 rok. Díky tomu lze aplikace testovat na reálných zařízeních, ale hlavně distribuovat pomocí obchodu Apple Store. Všechny aplikace před

jejich vydáním prochází přísným schvalovacím procesem, kdy jsou automaticky i ručně kontrolovány zda splňují všechny předepsané podmínky společnosti Apple.

Pro samotný vývoj jsou dále potřeba následující 3 věci:

- Počítač Mac se systémem Mac OS nebo OS X 10.8 (Mountain Lion) či novější,
- vývojové prostředí Xcode,
- iOS SDK ², které je součástí Xcode.

2.4 Hydroponické a závlahové systémy

Jednou z oblastí využití internetu věcí jsou inteligentní domácnosti, do jejichž podmnožiny patří závlahové systémy. Primárním účelem těchto systémů je zvýšit uživatelský komfort, snížit náklady na vodu, energie, a tím šetřit životní prostředí. [10]

Řídicím prvkem zahradních závlahových systémů jsou zahradní počítače. Ty se vyvinuly z časově spínaných relé až do současných chytrých zařízení připojených k internetu. Moderní zahradní počítače ke svému řízení využívají například předpovědi počasí, aby nespouštěly závlahu před deštěm, ale také množství různých senzorů. Mezi ty patří především dešťové čidlo, které v případě deště přeruší závlahu, větrné čidlo, senzory půdní vlhkosti či senzory teploty a osvětlení.

Trendem poslední doby se stává domácí pěstování rostlin, nebo také anglicky Urban Agriculture. Jedná o pěstování ovoce, zeleniny a bylinek v domácnostech, na balkónech či střeších domů. K tomu se mnohdy využívá právě moderních technologií, aby byly zaručeny optimální podmínky a pěstování tak bylo možné po celý rok. Na střeších domů či větších balkónech mohou nalézt uplatnění klasické chytré zahradní počítače, které byly zmíněny dříve. [21]

Účinným způsobem pěstování rostlin je tzv. Hydroponie. Hydroponie je pěstování rostlin bez půdy v živném roztoku. Rostliny jsou zasazeny do vhodného substrátu, nejčastěji keremzitu (expandovaný jíl), do vody jsou přidávány potřebné živiny v podobě hnojiv, regulováno její pH a teplota. Takto obohacená voda buď cirkuluje přes substrát, čímž jej zvlhčuje, nebo jsou kořeny rostlin ponořeny přímo do roztoku, který je probubláván za účelem obohacení kyslíkem. Výhodou hydroponie je rychlejší pěstování rostlin, kontrola nad přesnou skladbou živin ve vodě a především čistějším způsobem pěstování, což může být vhodné například pro alergiky. Hydroponicky vypěstované rostliny jsou také zdravější, než ty pěstované v půdě, jelikož nedochází ke styku s pesticidy a bakteriemi obsaženými v hlíně. [18] Moderní chytré závlahové systémy a kompletní hydroponické pěstební boxy se starají právě o správné nastavení všech těchto faktorů. Pěstební boxy jsou navíc zpravidla doplněny o umělé osvětlení, cirkulaci vzduchu a ohřev či chlazení vnitřního klimatu, čímž simulují rostlinám optimální prostředí, kterého by jinak v domácích podmínkách nebylo možné dosáhnout.

2.5 Přehled zařízení na trhu

V současné době se na trhu vyskytuje již řada chytrých zahradních počítačů, vyspělých pěstebních boxů či jiných chytrých zařízení usnadňující domácí pěstování rostlin. Ne všechny splňují parametry pro internet věcí, nicméně i tak slouží jako inspirace pro tuto práci.

²Software Development Kit

Rachio

Rachio je zástupcem chytrých zahradních počítačů, který existuje již ve dvou generacích. Umožňuje jednoduše nahradit téměř libovolný současný zahradní počítač. Jeho hlavní předností je úspora vody a energií. Dle výrobce dokáže ušetřit až 50 % spotřebované vody, díky inteligentnímu systému zalévání.



Obrázek 2.3: Rachio 2nd Generation (zdroj: Amazon)

Základní vlastnosti:³

- Podporuje 8 nebo 16 zalévacích zón, kdy pro každou zónu je možné nastavit několik parametrů, na základě kterých se automaticky optimalizuje zalévací cyklus,
- je připojen k internetu skrze Wi-Fi, poskytuje mobilní aplikaci pro nastavení a monitorování systému,
- využívá lokální předpovědi počasí (nezalévá při dešti, chladném počasí, nebo špatných povětrnostních podmínkách),
- umožňuje připojení externích senzorů jiných výrobců, včetně bezdrátového připojení,
- spolupracuje s Amazon Echo (umožňuje hlasové ovládání), nebo s chytrými zařízeními od společnosti Nest Labs.

Niwa

Niwa je chytrý a zároveň velmi hezky designově udělaný pěstební box. Vznikl jako startup za pomoci serveru kickstarter.com, kde jej podpořilo více než 500 lidí. Ačkoliv se jedná o projekt z roku 2014, stále je ve fázi předobjednávek.

Základní vlastnosti:⁴

³Čerpáno z <http://rachio.com>

⁴Čerpáno z <http://getniwa.com>



Obrázek 2.4: Niwa One (zdroj: Niwa)

- Hydroponický systém s automatickým zaléváním,
- je připojen do Cloudu, k nastavení a monitorování poskytuje mobilní aplikaci,
- obsahuje řadu přednastavených programů pro různé druhy rostlin (stačí pouze zasadit semínko a zvolit program),
- automaticky udržuje vhodné klima a osvětlení uvnitř boxu,
- vyniká hezkým designem, ve verzi Premium obsahuje hliníkové tělo.

Leaf

Další ze série pěstebních boxů je Leaf. Tento systém má velmi specifické využití a to pro pěstování léčebné marihuany. Je zde ovšem možné pěstovat i jiné druhy rostlin. Jedná se opět o hydroponický systém, obsahuje velké množství senzorů, včetně senzoru výšky rostlin. Automaticky provádí dávkování několika druhů hnojiv, je vybaven HD kamerou, chytrým systémem LED osvětlení, vzduchotechnikou, karbonovými filtry, nebo třeba vestavěným boxem se zahradním nářadím pro údržbu rostlin. Celý systém je připojen do Cloudu a pro správu nabízí mobilní aplikaci. Leaf také vytváří komunitu uživatelů, kteří mohou skrze aplikaci sdílet svoje nastavení pro pěstování, záznamy kamery a další informace. Aplikace nabízí video návody jak se o rostliny starat, stejně jako poskytuje denní hlášení, grafy a časosběrná videa z kamery.⁵

⁵Čerpáno z <https://getleaf.co>

LEAF

CLICK AND GROW™



Obrázek 2.5: vlevo Leaf (zdroj: Leaf), vpravo Click and Grow – Smart Garden (zdroj: Click and Grow)

Click and Grow

Click and Grow nabízí chytré řešení pro pěstování bylinek, zeleniny a jiných květin v domácích podmínkách po celý rok. Rostliny nejsou pěstovány hydroponicky, ale v tzv. chytré hlíně (smart soil), kterou si společnost vyvinula. Hlína obsahuje všechny potřebné živiny, které uvolňuje postupně během růstu rostlin, udržuje stále vhodné pH a rovnoměrně rozvádí vodu a kyslík ke kořenům. Semínka rostlin jsou prodávána zvláště, již připravené v této speciální hlíně. Pro pěstování jsou nabízeny dva druhy produktů. Prvním z nich je Smart Garden, který lze vidět na obrázku 2.5 vpravo. Tento chytrý květináč se nedá řadit mezi internet věci, jelikož není připojen do sítě internetu. Obsahuje LED osvětlení s fixním časovačem, který po 16 hodin denně poskytuje rostlinám světlo a zásobník na vodu (vystačí až na jeden měsíc). Celý systém vyniká hezkým designem, jenž je možné sladit se zbytkem domácnosti pomocí násuvných krytů na spodní část truhlíku. Druhým produktem je Wall Farm – systém polic, ve kterých lze pěstovat až 51 rostlin současně. Tento systém lze již připojit k internetu a poskytuje také mobilní aplikaci.⁶

Parrot Flower Power

Senzor do květináčů od společnosti Parrot nazvaný Flower Power umožňuje jednoduše monitorovat domácí květiny. Stačí jej zapichnout do libovolného květináče a přes mobilní aplikaci nastavit o jakou rostlinu se jedná. Aplikace nabízí na výběr ze 7000 různých rostlin, kdy pro každou z nich obsahuje specifické požadavky pro její správný růst. Flower Power je vybaven senzory slunečního světla, teploty, vlhkosti půdy a kvality půdy. S mobilní aplikací je připojen přes Bluetooth 4.0. Aplikace na základě zvoleného druhu rostliny a údajů ze senzorů informuje uživatele o nutných úkonech pro správný růst. Nabízí možnost propojení s IFTTT (*If This Then That*), díky čemuž je možné nastavit externí události, které se mají

⁶Čerpáno z <https://www.clickandgrow.com>

vykonat. Například uložení úkolu mezi poznámky. Senzor je vodotěsný a vše je napájeno pomocí AAA baterie, která zaručuje výdrž až 6 měsíců. Alternativou, která dokáže rostliny sama zalévat, je Flower Power H2O a chytrý květináč Parrot Pot. Oba produkty byly představeny na veletrhu CES 2015 a nejsou zatím na trhu dostupné.⁷



Obrázek 2.6: vlevo Parrot Flower Power (zdroj: Parrot), vpravo Xiami Mi Plant (zdroj: Xiaoland)

Xiami Mi Plant

Čínská společnost Xiami přišla se svým vlastním chytrým senzorem do květináčů. Nabízí stejné senzory i způsob komunikace, jako výše uvedený Flower Power. Oproti němu, ale vyniká podstatně nižší cenou a údajně větším výdržem baterie (až 1 rok). Celý senzor je také vodotěsný s certifikací IP5 (chráněno proti stříkající vodě). Mobilní aplikace obsahuje databázi 3tis různých rostlin a naměřené data ukládá do Cloudu.⁸

⁷Čerpáno z <http://global.parrot.com/au/products/flower-power/>

⁸Čerpáno z <http://xiaomi-mi.com/smart-home/xiaomi-huahuacacao-flower-care-smart-monitor/>

Kapitola 3

Bezdrátová komunikace a dostupný hardware pro IoT

Aby bylo možné připojit fyzické zařízení do internetu věcí, je nutné mít specifikovaný způsob adresování těchto zařízení, technologii pro jejich komunikaci a také jednotné komunikační protokoly. Ačkoliv internet věcí zatím nemá určené přesné specifikace, existuje v dnešní době několik protokolů, které jsou využívány a jeví se jako správné. Tato kapitola se věnuje představením právě těchto komunikačních protokolů, současným standardům pro bezdrátovou komunikaci senzorů a modulům pro jejich realizaci. Dále popisuje jednodeskové počítače, senzory a akční prvky vhodné pro stavbu chytrých zařízení.

3.1 Komunikační protokoly

Protokol je sada pravidel pro komunikaci mezi počítači. Obsahuje pravidla, jak zahájit komunikaci, formát výměny zpráv a způsob ukončení komunikaci. Určuje, jakým vstupům komunikující zařízení rozumí a jaký bude výstup. Také specifikuje, jak budou zprávy posílány, bezpečnost a způsob zpracování chyb. [13]

HTTP/HTTP2

Nejpoužívanějším komunikačním protokolem pro web na internetu je HTTP. Verze 1.1 je definována v RFC 2616. Jedná se o textový protokol určený pro výměnu hypertextových dokumentů ve formátu HTML. Pracuje nad TCP/IP, nejčastěji na portu 80 a funguje způsobem dotaz – odpověď. Pro účely internetu věcí připojené zařízení vystupuje v roli serveru, který poskytuje odpovědi klientovi, nejčastěji v podobě webového prohlížeče, nebo mobilní aplikaci napojené skrze architekturu REST. Hodnoty předávané mezi klientem a serverem jsou adresovány pomocí URI¹, stejně jako v případě klasických webových stránek. K dotazování slouží HTTP metody GET, POST, PUT a DELETE. Pro získání například teploty ze senzoru s ID 1 se může volat metoda GET nad URL `/sensors/1/temperature`. [9, 16] Tento protokol se nejeví jako vhodný pro použití v internetu věcí. Je bezstavový a pro každý dotaz vyžaduje nové navázání spojení, které obsahuje složité hlavičky. Režie na komunikaci je tak příliš veliká, což má dopad hlavně na bateriově napájené zařízení.

Nová verze protokolu HTTP/2 byla standardizována v roce 2015. Narozdíl od verze 1.1 dovoluje přenášet mnoho požadavků současně skrze jediné otevřené TCP spojení (tzv.

¹Uniform Resource Identifier

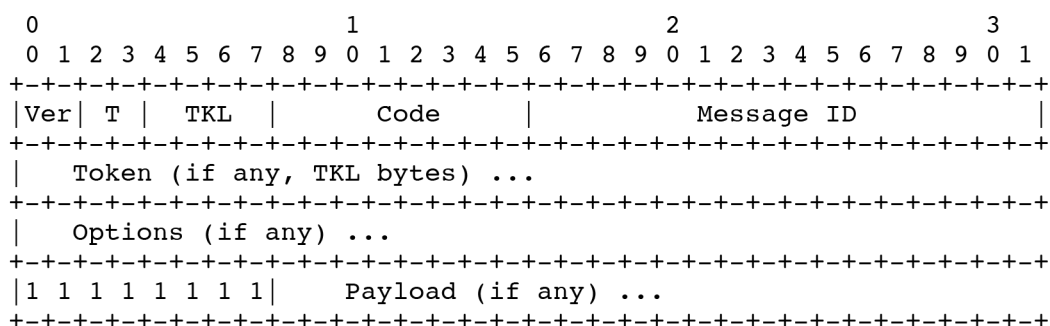
Multiplexování), HTTP hlavičky jsou komprimovány a data se přenáší v binárním formátu. Tím dochází ke zvýšení rychlosti přenosu a snížení latence, díky čemuž je tento protokol již lepším kandidátem pro použití v internetu věcí. [7]

CoAP

Constrained Application Protocol (CoAP), definovaný v RFC 7252, je protokol určený pro komunikaci mezi zařízeními (M2M², IoT) s omezenými zdroji (především s důrazem na nízkou spotřebu energie). Pro komunikaci využívá protokol UDP. Není zde tedy primárně zaručen spolehlivý přenos dat, jelikož samotný UDP protokol nezaručuje správné doručení datagramů. Na druhou stranu velikost UDP datagramů je podstatně menší než u TCP.

Protokol vyniká svou jednoduchou integrovatelností s HTTP. Pro vývojáře se tak na venek tváří naprosto stejně. Pracuje se známými HTTP metodami GET, POST, PUT a DELETE. Podporuje architekturu REST a komunikuje na principu dotaz – odpověď. Dále podporuje unicast a multicast, adresování pomocí URI, content-type pro různé typy přenášených zpráv, proxy a cachování. Narozdíl od HTTP má nízkou komunikační režii, je celkově jednodušší, pracuje asynchronně a je v binárním formátu (stejně jako HTTP/2). Protokol je primárně určen pro přímou komunikaci mezi dvěma zařízeními (one-to-one), ovšem díky podpoře multicasu je možné adresovat více zařízení současně. Pro zajištění bezpečnosti se využívá DTLS (Datagram Transport Layer Security) protokol. Ten je alternativou k TLS pro nestavové datagramové protokoly (např. UDP). Tím je umožněno zabránit odposlechům, falšování, nebo padělání zpráv.

CoAP definuje čtyři typy zpráv – *Confirmable*, *Non-confirmable*, *Acknowledgement* a *Reset*. Každá zpráva obsahuje *Message ID*, které slouží k detekci duplicit a pro případnou spolehlivost. Podle toho, zda je potřeba zaručit spolehlivost doručení, rozlišují se dva typy komunikace. První, která vyžaduje spolehlivost, značí odchozí zprávu jako *Confirmable (CON)* spolu s jejím *Message ID*. Příjemce potom musí odpovědět zprávou *Acknowledgement (ACK)* před vypršením nastaveného časovače (time out). Pokud příjemce není schopen zprávu přijmout, odpovídá zprávou *Reset (RST)* namísto *Acknowledgement (ACK)*. Druhý typ komunikace, kde správné doručení není důležité (např. naměřená data ze senzorů), využívá zpráv *Non-confirmable (NON)* a *Message ID* pro detekci duplicit. [19]



Obrázek 3.1: Formát zprávy CoAP (zdroj: IETF)

²Machine to Machine

MQTT

MQ Telemetry Transport (MQTT) je jednoduchý a nenáročný protokol představený v roce 1999 společností IBM. V roce 2013 proběhla jeho standardizace OASIS. Byl navržen zejména pro zařízení s nízkým výkonem a díky tomu se relativně rychle rozšířil. Své využití tak nachází v senzorech a chytrých zařízeních pro internet věcí. Protokol umožňuje předávání zpráv mezi klienty prostřednictvím centrálního bodu tzv. *brokeru*. Přenos probíhá pomocí TCP a používá model *publish – subscribe*.

Zařízení (publisher) odesílá data do centrálního bodu – brokeru, který tyto data následně poskytuje odběratelům (subscriber). Zprávy jsou tříděny do tzv. témat (topics). Zařízení buď publikuje data v daném tématu, nebo je přihlášeno k odběru tématu či témat a broker pak všechny zprávy s daným tématem posílá do zařízení. Jedno zařízení může v jeden okamžik vystupovat v obou rolích. Tedy publikuje do některých témat a zároveň odebírá data z témat jiných. Témata mají hierarchickou strukturu, kterou lze vidět na obrázku 3.2. Jednotlivé úrovně jsou identifikovány pomocí textových řetězců a odděleny znakem „/“.

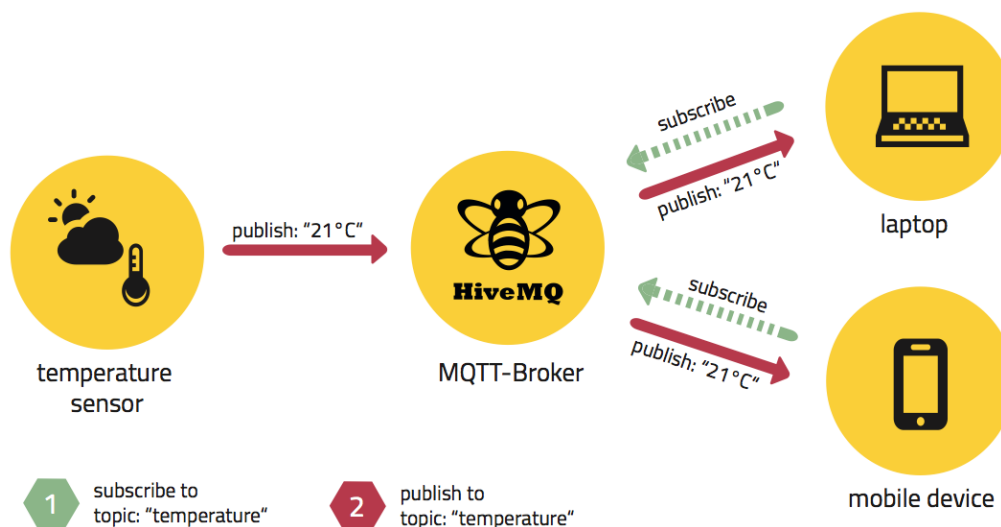


Obrázek 3.2: Hierarchická struktura témat v MQTT (zdroj: HiveMQ.com)

K navázání spojení mezi klientem (zařízením) a brokerem slouží zpráva `CONNECT`. Spolu s ní je možné předat příznak *clean session*, který smaže všechny staré odběry daného klienta. Broker odpoví zprávou `CONNACK` a klient se následně může následně přihlašovat (`SUBSCRIBE`) a odhlašovat (`UNSUBSCRIBE`) k odběru jednotlivým témat. Klient se přihlašuje k odběru konkrétního tématu, nebo může využít zástupných znaků „#“ a „+“ k odběru celé skupiny témat najednou. Znak plus nahrazuje jednu úroveň v hierarchii. Např. v příkladu uvedeném na obrázku 3.2, lze nahradit úroveň „livingroom“ zástupným znakem „+“ a tím odebírat témata „temperature“ ve všech místnostech v přízemí (úroveň „groundfloor“). Znak mřížka nahrazuje jednu či více úrovní a musí být vždy jako poslední. Přihlášení k odběru tématu „myhome/groundfloor/#“ v uvedeném příkladu znamená, že budou přijímány zprávy týkající se všech témat v přízemí domu. Témata začínající znakem „\$“ jsou speciální, používaná brokerem. Samotný obsah posílaných zpráv není nikterak omezen. Jedná se o binární data s maximální velikostí 256MB. Nejčastěji se používá formát JSON, XML, nebo textové zprávy.

MQTT definuje tři úrovně spolehlivosti doručení zpráv (QoS – Quality of Service):

- **QoS 0: At most once delivery** – nejnižší úroveň, někdy označovaná také jako „Fire and Forget“. Poskytovatel (publisher) pouze pošle zprávu `PUBLISH` brokeru a ten jej přepošle odběratelům daného tématu bez jakékoliv záruky doručení.
- **QoS 1: At least once delivery** – poskytovatel odešle zprávu `PUBLISH` a čeká na potvrzení od brokeru. Ten zprávy přepošle odběratelům se stejnou nebo nižší úrovní QoS a jakmile dostane potvrzení `PUBACK`, přepošle jej poskytovateli. Broker může potvrzení odeslat aniž by měl příjem zprávy potvrzený od všech příjemců.



Obrázek 3.3: MQTT publish – subscribe model (zdroj: HiveMQ.com)

- **QoS 2: Exactly once delivery** – jedná se o nejvyšší stupeň QoS, kde se neakceptuje ztráta ani duplikace zprávy. Poskytovatel pošle brokeru zprávu PUBLISH s QoS nastavenou na 2. Ten ji přepošle odběratelům a odpoví zprávou PUBREC (potvrzení přijetí). Poskytovatel odpoví zprávou PUBREL, broker smaže zprávu a odpoví zprávou PUBCOMP. Tím se komunikace uzavře. Tato úroveň QoS se používá pro kritické operace, jako vypnutí/zapnutí akčních prvků.

K zajištění bezpečnosti je doporučeno používat šifrování zpráv pomocí TLS/SSL. Zároveň při připojování klienta k brokeru se provádí autorizace pomocí uživatelského jména a hesla. [6]

Spolu s CoAP se jedná v dnešní době o nejrozšířenější protokol používaný pro internet věcí a M2M komunikaci. Je přímo podporován v řadě cloudových služeb a existuje spousta implementací brokeru a knihoven pro komunikaci.

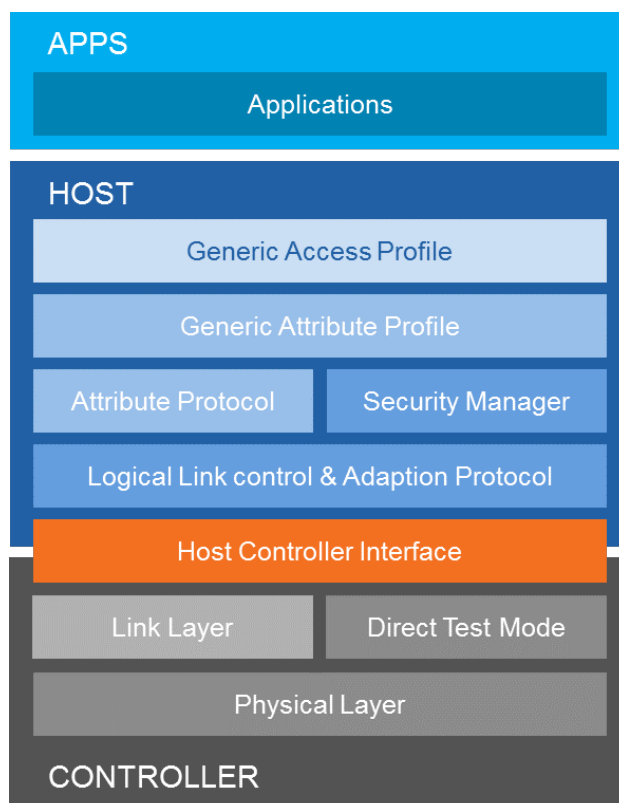
MQTT-SN

Jedná se o zjednodušenou verzi protokolu MQTT nazvanou MQTT for Sensor Networks. Jak název napovídá, tento protokol je optimalizován pro použití v rádiově propojených sensorových sítích (např. přes ZigBee). Je zde optimalizována komunikace, aby se přenášelo co nejméně dat. [20] Bohužel se tento protokol od poslední specifikace v roce 2013 dále nerozvíjí a jeho rozšíření není příliš veliké.

3.2 Standardy pro bezdrátovou komunikaci

Bluetooth

Standard pro bezdrátovou komunikaci Bluetooth byl vytvořen v roce 1994 jako bezdrátová alternativa pro sériové rozhraní RS-232. Slouží k budování soukromých bezdrátových sítí (PAN). Bluetooth je definován standardem IEEE 802.15.1 a vyskytuje se v několika verzích.



Obrázek 3.4: Bluetooth Stack (zdroj: Bluetooth.com)

Jednotlivé verze se od sebe liší přenosovou rychlostí, propustností a dosahem. Rozhraním pro Bluetooth komunikaci je vybavena velká spousta elektronických zařízení jako mobilní telefony, nositelná elektronika, notebooky, automobily, televize a jiné, čímž jej tvoří jedním z nejpožívanějších způsobů bezdrátové komunikace.

Bluetooth pracuje v bezlicenčním pásmu na frekvencích 2402 až 2480 MHz rozděleném do 79 kanálů o šířce 1Mhz. K přenosu se využívá metody FHSS (*Frequency Hopping Spread Spectrum*), tedy přeskakování mezi několika frekvencemi při přenosu bitů za účelem zvýšení kvality přenosu. Bluetooth zařízení jsou rozdělena do tří tříd podle výkonnosti. S nižším výkonem se snižuje maximální dosah, ale také energetická náročnost. V mobilních telefonech se nejčastěji používá třída 2, která má maximální povolený výkon 2,5mW a dosah 5-10m. Maximální dosah je u Bluetooth 5 až 400m ve volném prostoru. V reálném prostředí, kde je přenos blokován například zdí, dosah velmi rapidně klesá. Bluetooth využívá master/slave modelu, kdy jedno zařízení vystupuje jako master a může k němu být připojeno až 7 komunikujících zařízení slave. Takto definovaná síť se nazývá *piconet*. Jedno slave zařízení může být připojeno vždy jen k jednomu master zařízení, není tak umožněna přímá komunikace mezi slave zařízeními. Adresování probíhá pomocí 48-bitové adresy, častěji je zařízením přiřazeno jméno, které lze libovolně editovat a zobrazuje se ostatním zařízením při párování.

Bluetooth 5 byl schválen v prosinci 2016 a přináší řadu zásadních novinek. Mezi hlavní patří zvýšení dosahu až na čtyřnásobek oproti verzi 4.2 (až 400m) a zrychlení komunikace na dvojnásobek tedy 2 Mbps. [1] Novou verzí se Bluetooth snaží prosadit právě v oblasti internetu věcí, kde v poslední době vznikají jiné alternativy jako třeba Zigbee. Jelikož všechny

Verze protokolu	Rok vydání	Frekvence [GHz]	Šířka pásma [MHz]	Maximální rychlost [Mbit/s]	Způsob modulace	Dosah [m]
802.11	1997	2,4	22	2	DSSS, FHSS	100
802.11a	1999	5	20	54	OFDM	120
802.11b	1999	2,4	22	11	DSSS	140
802.11g	2003	2,4	20	54	OFDM	140
802.11n	2009	2,4/5	20/40	150	MIMO-OFDM	250
802.11ac	2013	2,4 a 5	až 160	866	MIMO-OFDM	250

Tabulka 3.1: Přehled standardů IEEE 802.11

předchozí verze byly vždy velmi rychle přijaty výrobci elektroniky, je pravděpodobné, že rozšíření Bluetooth 5 bude relativně rychlé.

Bluetooth Low Energy

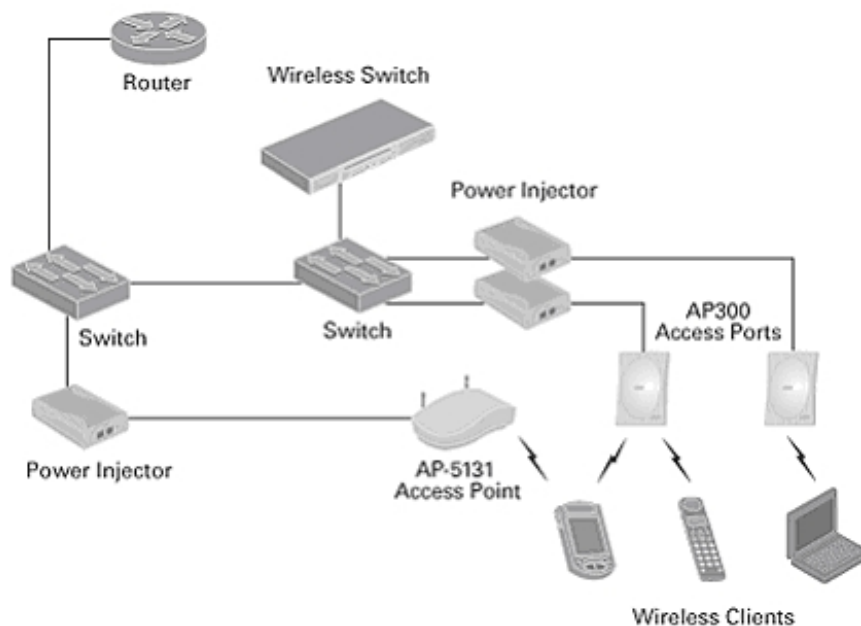
Bluetooth Low Energy (BLE), někdy také označované jako Bluetooth Smart, je nízkoenergetická verze Bluetooth určená pro internet věcí, zejména vhodná pro bateriově napájené senzory. Většina zařízení na trhu, které podporují klasický Bluetooth podporuje i BLE. Obě technologie tak fungují vedle sebe a každá z nich má jiný účel využití.

Snížení spotřeby nelze realizovat snižováním vysílacího výkonu, jelikož tím se zkracuje také dosah. Využívá se proto zkrácení na minimum doby, ve které se modul probudí, naváže spojení, provede komunikaci a opět přejde do úsporného režimu. Tato doba se nazývá „enumeration time“ a je zhruba 20x nižší než u klasického Bluetooth. Na druhou stranu je zde nižší i maximální přenosová rychlost. [15, 1] Jelikož se ale zpravidla nepřenáší velké množství dat ze senzorů, není tento parametr tak zásadní.

Wi-Fi

Technologie Wi-Fi je označení pro několik standardů IEEE 802.11x, kde x označuje jednotlivé verze. Používá se pro bezdrátovou komunikaci v lokální síti (WLAN) využívající elektromagnetických rádiových vln v bezlicenčním pásmu 2,4 GHz a 5 GHz. Původní standard byl představen již v roce 1997 a postupně k němu byly definovány dodatky s označením *a*, *b*, *g* a *n*. Standardy 802.11b a 802.11g využívají pásmo 2,4 GHz, které je díky tomu velmi zahlcené a může tak docházet k rušení s ostatními službami na této frekvenci (např. Bluetooth). Verze 802.11a pracuje na frekvenci 5 GHz, stejně jako standard 802.11n, který podporuje obě zmíněné frekvence. Nový standard 802.11ac nabízí tzv. dual-band, tedy využití obou pásem současně. Přehled nejpoužívanějších verzí je uveden v tabulce 3.1. O dodržování standardu se stará Wi-Fi Alliance. Ta po splnění podmínek přiděluje výrobci zařízení logo, které ujišťuje uživatele, že zařízení je schopné komunikovat s ostatními pomocí Wi-Fi technologie. [2]

Základem bezdrátové sítě je přístupový bod (AP – Access Point). Ten vytváří bezdrátovou síť identifikovatelnou pomocí SSID (Service Set Identifier) – jedná se o řetězec maximálně 32 ASCII znaků. Wi-Fi zařízení se připojují do této sítě a komunikují skrze přístupový bod. Vyjimku tvoří pouze tzv. ad-hoc sítě, kdy zařízení mohou komunikovat napřímo. Každé zařízení v síti je rozlišeno pomocí IP adresy (IPv4, nebo IPv6). K připojení domácí lokální sítě do internetu slouží router, který v sobě často kombinuje funkci



Obrázek 3.5: Architektura Wi-Fi sítě (zdroj: Eprin.cz)

klasického routeru a přístupového bodu (*Wi-Fi routery*). Ukázkou architektury Wi-Fi sítě lze vidět na obrázku 3.5.

Díky svojí rozšířenosti, standardizaci a fungujícím způsobu zabezpečení si Wi-Fi slibuje široké využití v internetu věcí. Jeho nevýhodou je pouze vyšší energetická náročnost.

Wi-Fi Direct

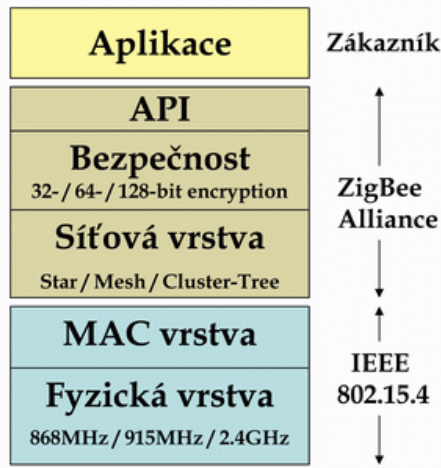
Wi-Fi Direct je standard Wi-Fi, který dovoluje přímé spojení zařízení bez nutnosti přístupového bodu (AP). Využívá se v mobilních telefonech, kamerách, tiskárnách a dalších zařízeních k přímému sdílení například fotografií, přenosu dat apod. Zařízení mohou tvořit *one-to-one* spojení, nebo může být připojeno několik zařízení současně. K navázání spojení stačí aby pouze jedno zařízení podporovalo Wi-Fi Direct. Párování je realizováno stisknutím tlačítka na zařízeních, pomocí NFC³ technologie nebo zadáním PIN kódu. K zabezpečení se využívá technologie WPA2. Wi-Fi Direct nevyžaduje připojení k internetu a je jistou alternativou k technologii Bluetooth. [3]

ZigBee

ZigBee je bezdrátová komunikační technologie označená standardem IEEE 802.15.4, která umožňuje vzájemnou komunikaci mnoha zařízení na vzdálenost několika stovek metrů v sítích PAN⁴. Jedná se o relativně nový standard z roku 2004, který je spravován organizací ZigBee Alliance. Ta sdružuje řadu velkých firem zabývajících se domácí automatizací a internetem věcí jako Schneider Electric, Texas Instruments, Cisco, Samsung, nebo Honeywell. [4] Díky nízkým nárokům na hardware a nízké spotřebě nachází uplatnění v oblastech automatizace budov, průmyslové automatizace, spotřební elektroniky, počítačových periferií

³Near Field Communication

⁴Personal Area Network



Obrázek 3.6: OSI model protokolu ZigBee (zdroj: vyvoj.hw.cz)

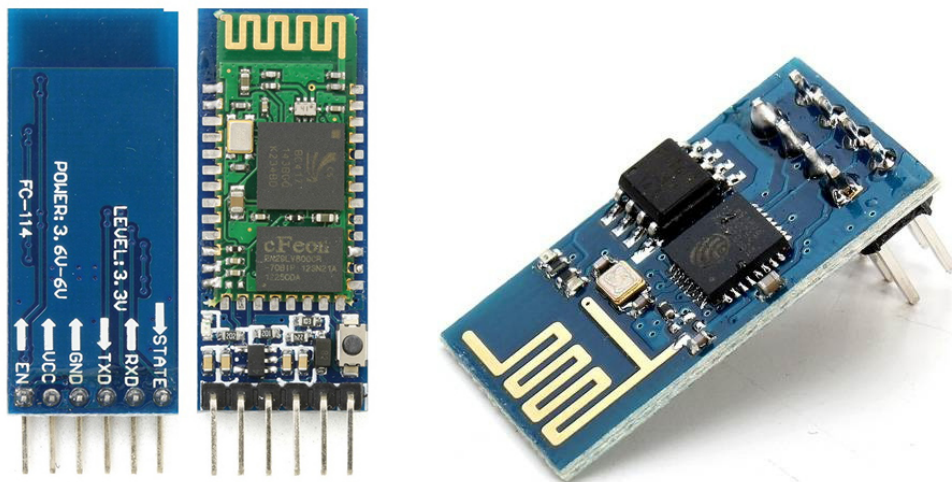
a zdravotnictví. Zigbee má zaplnit mezeru mezi technologiemi Wi-Fi a Bluetooth, které nemusí být pro určité aplikace vždy vhodné. Bluetooth se tomu snaží předejít nově představenou verzí Bluetooth 5.

ZigBee pracuje v bezlicenčních pásmech 2,4 GHz, 915 MHz a 858 MHz. Nízká spotřeba je vykoupena nižší rychlostí přenosu, která však v mnoha aplikacích plně postačuje. Díky tomu je možné provozovat bateriově napájené zařízení po dobu 100 až 1000 dnů. Nižší přenosová rychlost poskytuje také vyšší odolnost proti rušení, čímž je protokol vhodnější pro použití v průmyslu.

Model protokolu lze rozdělit do tří hlavních vrstev, které lze vidět na obrázku 3.6. **Fyzická vrstva** spolu s linkovou definuje způsob bezdrátové komunikace, výše uvedené používané frekvenční pásma a komunikaci mezi jednotlivými zařízeními prostřednictvím rámců. **Síťová vrstva** definuje 3 základní topologie sítě – hvězda (star), strom (tree) a síť (mesh), jenž je kombinací dvou předchozích. Dále zajišťuje připojování k síti a odpojování od ní, zabezpečení a směrování paketů. Jako základní zabezpečovací mechanismus se používá kryptografický algoritmus AES (Advanced Encryption Standard). Poslední **aplikační vrstva** protokolu se skládá z pomocné aplikační podvrstvy (APS), objektů ZigBee (ZDO) a uživatelských aplikačních objektů. Aplikační pomocná podvrstva umožňuje párování zařízení podle poskytovaných služeb a požadavků. Objekt ZigBee definuje roli zařízení v rámci sítě a spravuje poskytované služby. Objekt může vystupovat ve třech různých rolích:

- **Koordinátor** – řídí celou síť a v topologii je právě jeden. Shromažďuje informace o síti a stará se o zabezpečení.
- **Zařízení s plnou funkčností (FFD)** – implementuje kompletní protokolový rámec a zajišťuje veškeré služby, které ZigBee stanovuje. V topologii slouží jako směrovač, který přeposílá data v rámci sítě. Koordinátor je také FFD.
- **Zařízení s omezenou funkčností (RFD)** – implementuje jen nezbytné protokolové knihovny za účelem snížení hardwarové náročnosti. Tato zařízení mohou pracovat pouze jako koncová a komunikují s koordinátorem nebo směrovačem.

[17]



Obrázek 3.7: vlevo HC-05 modul, vpravo esp8266 (zdroj: santy.cz)

3.3 Komunikační moduly

HC-05/HC-06

Jedná se o oblíbený bluetooth modul, který ke své činnosti potřebuje pouze napájecí napětí 3,3V. HC-05 podporuje jak režim MASTER, tak i SLAVE, zatímco HC-06 pouze režim SLAVE a může tak komunikovat jen s jinými zařízeními, které vystupují v roli MASTER (např. mobilní telefon, notebook, nebo HC-05 modul). Modul se může nacházet v jednom ze dvou módů – programovacím pro nastavení a provozním, kdy vysílá a přijímá data. Nastavení v programovacím režimu probíhá pomocí AT příkazů. Modul se většinou prodává napájený na rozšiřující destičce, která umožňuje jeho snadnější připojení, použití 5V logiky a poskytuje signalizační LED. Udávaný dosah je až 10m a anténa je integrována přímo v modulu.

HM-10 BLE 4.0

Bluetooth low energy modul HM-10 je nástupcem modulu HC-05. Podporuje Bluetooth 4.0 a BLE, zatímco HC-05 pouze starší specifikaci 2.0.

Popis pinů (stejný i pro HC-05 a HC-06):

- STATE – určuje stav modulu,
- VCC – napájení,
- GND – uzemnění,
- TXD – sériové rozhraní pro vysílání dat,
- RXD – sériové rozhraní pro příjem dat,
- KEY – slouží pro přepnutí mezi programovacím a provozním režimem.

ESP8266

ESP8266 je asi nejrozšířenější a nejlevnější Wi-Fi modul na trhu. K modulu je díky široké komunitě uživatelů dostupný dostatek materiálů, návodů i vlastního firmware. Kromě samotného Wi-Fi rozhraní obsahuje modul přímo řídicí mikrokontrolér, takže jej lze rovnou naprogramovat a využít pro jednoduché bezdrátové senzory. Programovat lze v jazycích Lua či MicroPython, existuje ale i způsob jak jej programovat skrze Arduino IDE v jazyce Arduino (podobný C/C++).

Modul pracuje s 3,3V logikou, podporuje Wi-Fi standardy 802.11b/g/n a zabezpečení pomocí WPA/WPA2. Existuje v několika variantách, které se liší typem antény, počtem vstup/výstupních (GPIO) pinů a tvarem desky. Základní a pravděpodobně nejpoužívanější variantu ESP-01 lze vidět na obrázku 3.7 vpravo. Tato varianta disponuje dvěma GPIO piny.

NRF24L01

Modul NRF24L01 je na první pohled velmi podobný modulu ESP8266. Narozdíl od něj neobsahuje mikrokontrolér a nelze se s ním připojit přímo k internetu. Zajišťuje pouze rádiovou komunikaci na frekvencích 2,4 až 2,525GHz mezi dvěma NRF24L01 moduly připojenými například k arduino deskám. Vysílací frekvenci lze zvolit s rozsahem 1MHz, čímž vzniká 125 různých vysílacích pásem. Jelikož Wi-Fi technologie využívá většinu spodních pásem, doporučuje se použití horních 25 frekvencí (2,5 až 2,525GHz). Modul se vyrábí v několika variantách podle typu antény. Vyžaduje napájecí napětí 3,3V a dokáže pracovat i s 5V logikou. Vysílací výkon lze nastavit ve čtyřech úrovních, ovšem pro vyšší úroveň je doporučeno použít externí napájení. Modul se vyznačuje velmi nízkou spotřebou v klidovém stavu (pouze $1\mu A$). Při odesílání a příjmu dat může krátkodobě odebírat mnohonásobně vyšší proud (až stovky μA). Proto je vhodné zapojení kondenzátoru mezi napětí a zem, pro pokrytí těchto výkyvů.

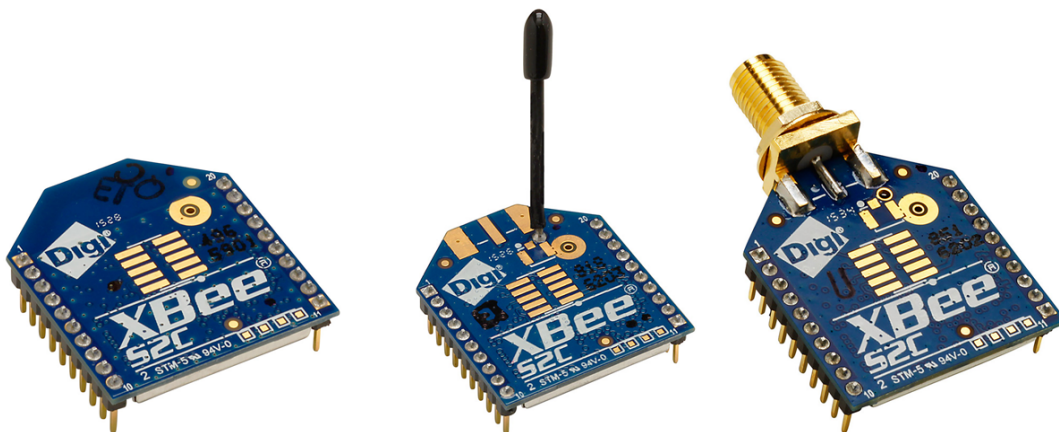
Komunikace s mikrokontrolerem probíhá pomocí SPI rozhraní. Obsahuje vestavěnou opravu chyb při odesílání a automatické znovu posílání zpráv. Modul dokáže současně komunikovat až s šesti dalšími zařízeními. Pro správu komunikace existuje řada hotových knihoven, z nichž nejznámější pochází od uživatele Tmrh20⁵. Od stejného uživatele pochází také knihovna RF24Mesh, která simuluje mesh síť vytvořenou pomocí NRF24L01 modulů.

XBee

Společnost Digi International nabízí několik druhů XBee modulů pro bezdrátovou komunikaci založenou na standardu 802.15.4 a ZigBee. Moduly nabízí jednoduchou implementaci, nízkou spotřebu a topologické volby typu *point-to-multipoint* a *mesh*.

Moduly se vyskytují ve dvou sériích (XBee série 1 a XBee série 2) a každá z těchto sérií ještě v normální a PRO verzi. XBee série 1 využívá mikročip firmy Freescale a poskytuje topologii typu hvězda (star) a point-to-point. Nativně tak nepodporuje topologii typu mesh, narozdíl od série 2, která využívá mikročip firmy Ember Networks. Moduly různé série spolu nemohou komunikovat. Verze PRO nabízí oproti klasické zejména vyšší vysílací výkon a tím větší dosah, za cenu vyšší energetické náročnosti. Obvyklý dosah je do 100m, u verze PRO v rámci jednotek km, nejvýkonější verze s externí anténou dosahuje ve volném prostoru dosah až 40km.

⁵Dostupné z: <https://tmrh20.github.io>



Obrázek 3.8: XBee moduly s různými typy antén (zdroj: digi.com)

Každý modul se vyrábí s několika různými druhy antén:

- **Drátová anténa** – jedná se o kus vodiče vystupujícího z těla modulu. Ve většině případů je tento typ antény nevhodnější. Nabízí všesměrové vyzařování a dosah je ve všech směrech téměř totožný.
- **Anténa na čipu** – plochý keramický čip zabudovaný v těle modulu. Čip je menší a odolnější, čímž je vhodný například pro nositelná zařízení, kde není externí anténa žádoucí. Vyzařovací úhel má tvar srdce, díky čemuž je signál v mnoha směrech oslabený.
- **PCB anténa** – byla představena s XBee PRO série 2. PCB anténa je vytištěna přímo na desce XBee modulu. Nabízí stejné vlastnosti jako anténa na čipu, ovšem s nižší pořizovací cenou.
- **U.FL konektor** – jedná se o malý konektor pro externí anténu. Tento typ se využívá v případě, kdy je samotný XBee modul stíněný (např. v kovové krabici) a anténa je vyvedena ven z pouzdra.
- **RPSMA konektor** – velmi podobný U.FL konektoru pouze větší se stejným případem využití.

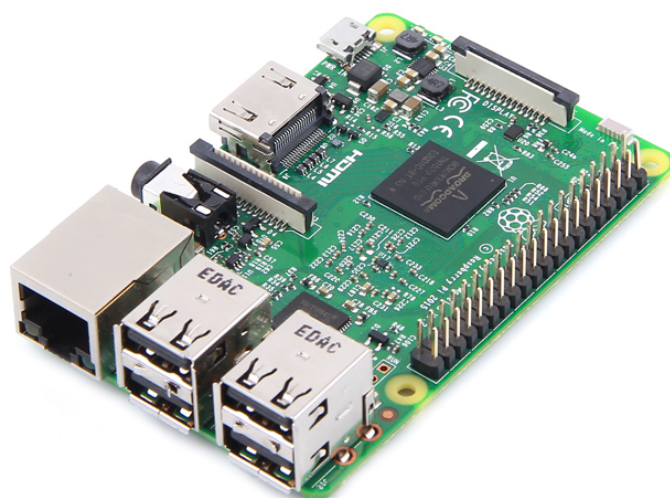
[8]

3.4 Jednodeskové počítače

Raspberry Pi

Raspberry Pi je jednodeskový počítač o velikosti kreditní karty. Poprvé byl představen v roce 2012 s cílem podpořit výuku informatiky na školách. Raspberry Pi je kompletní jednočipový počítač s operačním systémem Raspbian (systém odvozený od linuxové distribuce Debian, optimalizovaný pro ARM procesory). Je zde možné nainstalovat i jiné operační systémy, včetně odlehčené verze Microsoft Windows 10 IoT Core.

Existuje několik modelů Raspberry Pi, které se liší především svým výkonem a poskytovanou výbavou.



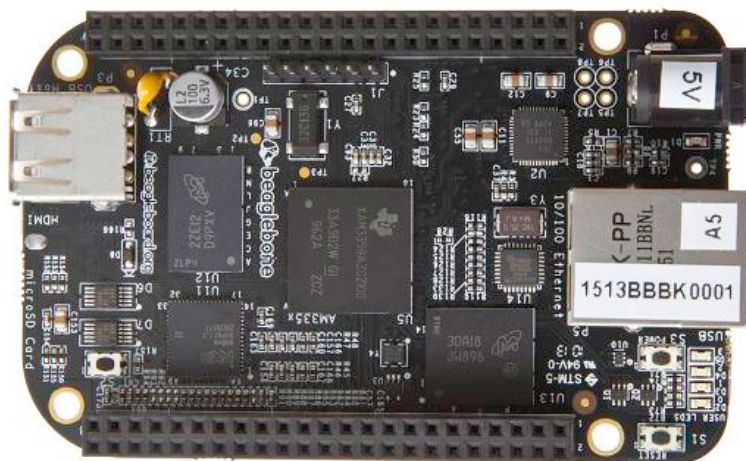
Obrázek 3.9: Raspberry Pi 3 Model B (zdroj: Raspberrypi.org)

- **Raspberry Pi 1 model A/B** – první verze z roku 2012 je vybavena jednojádrovým procesorem ARM, 256MB respektive 512MB operační paměti v případě modelu B a grafickým procesorem Broadcom VideoCore. Jako uložení slouží SD karty. Počítač poskytuje HDMI výstup, 1 (model A) nebo 2 (model B) USB porty, audio výstup, síťové rozhraní Ethernet, CSI konektor, DSI konektor, RCA konektor a 34 GPIO pinů.
- **Raspberry Pi 2 model B** – model z roku 2015 obsahuje již čtyřjádrový 32-bitový procesor ARM Cortex-A7, 1GB operační paměti a grafický procesor Broadcom VideoCore. Tato verze je vybavena MicroSDHC slotem, HDMI výstupem, audio výstupem, čtyřicet USB portů, síťovým rozhraním Ethernet s konektorem RJ25, CSI konektorem, DSI konektorem a 40 GPIO piny.
- **Raspberry Pi 3 model B** – nejnovější verze, která byla představena v roce 2016, je vybavena čtyř jádrovým 64-bitovým procesorem ARM Cortex-A53, 1GB operační paměti, grafickým procesorem Broadcom VideoCore IV a oproti svým předchůdcům integrovanými Wi-Fi a Bluetooth moduly, včetně podpory Bluetooth Low Energy. Tvarem ani nabídkou rozhraní se od svého předchůdce nijak neliší.
- **Raspberry Pi Zero** – tento model vystupuje z řady zejména svou velikostí, cenou a omezenými zdroji. Obsahuje pouze jednojádrový procesor ARM, 512MB operační paměti, Broadcom VideoCore grafický procesor, jeden Micro-USB konektor, Mini-HDMI rozhraní, MicroSDHC slot pro uložení a 40 neosazených GPIO pinů. Jeho cena je pouhých 5\$.

Pro připojení senzorů a akčních prvků k Raspberry Pi slouží **GPIO piny**. Verze 2 i 3 jich obsahuje celkem 40, z toho 17 pinů je standardně vstupně - výstupních, 2 piny slouží pro I^2C , 2 piny pro UART, 5 pinů pro SPI, dva 5V a dva 3,3V napájecí piny a zem. **CSI konektor** slouží pro připojení kamery a **DSI konektor** pro připojení externího displeje. K Raspberry se dodává řada originálních i neoriginálních kamer a dotykových displejů připojitelných právě skrz tyto rozhraní.

BeagleBone

Jednodeskový počítač BeagleBone je alternativou k výše uvedenému Raspberry Pi. Ve verzi BeagleBone Black nabízí procesor ARM Cortex-A8, 512MB DDR3 RAM, 3D grafický akcelerátor a integrovanou paměť s kapacitou 4GB. Oproti Raspberry Pi poskytuje pouze jeden USB port, síťové rozhraní Ethernet, HDMI a 2x 46 GPIO pinů. Jako základní operační systémy se nabízí Debian, Android, nebo Ubuntu.

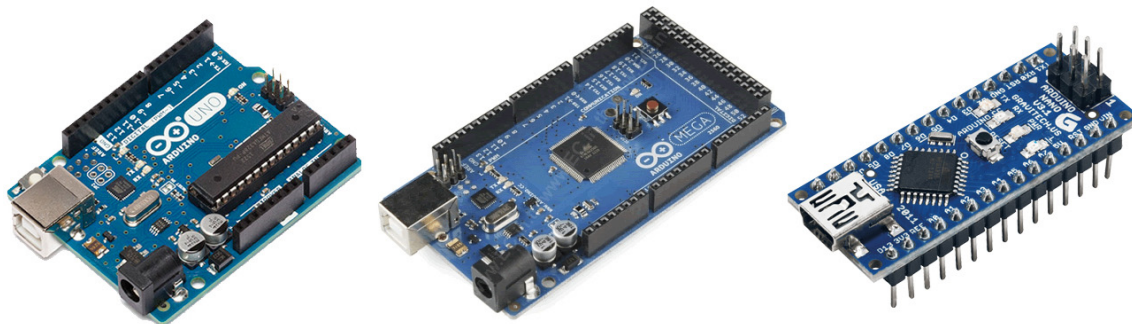


Obrázek 3.10: BeagleBone Black (zdroj: Beagleboard.org)

Arduino

Arduino je velmi oblíbená otevřená (*open source*) elektronická platforma, založená na uživatelsky jednoduchém hardware a software. Projekt jednodeskového počítače postaveného na mikrokontrolerech ATmega od firmy Atmel vznikl v roce 2005 v Itálii. Jeho cílem bylo vytvořit jednoduchou prototypovací platformu pro studenty. Velmi rychle však došlo k rozšíření a popularitě této platformy, díky čemuž se v současné době nabízí obrovské množství senzorů, akčních prvků, displejů a jiného příslušenství, stejně jako množství softwarových knihoven pro práci s nimi. Arduino narozdíl od Raspberry Pi není zamýšleno jako plnohodnotný počítač. Řídicí program je vyvíjen zvlášť a do Arduina posléze nahrán a spuštěn, kde běží nejčastěji v nekonečné smyčce. Celý projekt je od začátku volně dostupný (*open source*) všem uživatelům, kteří jej mohou vylepšovat. Díky volně dostupným schémátům desek Arduino se vyskytuje na trhu množství jejich neoriginálních kopií, které jsou ale svou funkčností srovnatelné s originálem.

Desky arduino obsahují 8-bitové mikrokontrolery ATmega od firmy Atmel a množství podpůrných obvodů. Existuje několik druhů desek, které se liší použitým mikrokontrolerem, velikostí paměti, množstvím I/O pinů, svou fyzickou velikostí a USB rozhraním. Nejznámější a nejpoužívanější jsou varianty Arduino Uno, Arduino Mega a Arduino Nano. Všechny tři typy desek lze vidět na obrázku 3.11. I/O piny slouží pro připojení vnějších senzorů a akčních prvků. Existuje také řada desek, tzv. *Shieldů*, které se nasadí na piny Arduina a poskytují tak jednodušší připojení senzorů, nebo třeba Ethernetové rozhraní apod. Arduino Uno disponuje celkem 14 digitálními I/O piny, 6 analogovými piny, jedním pinem pro reset,



Obrázek 3.11: zleva Arduino Uno, Arduino Mega a Arduino Nano (zdroj: Arduino.cc)

napájecími a zemičícími piny. Celkem 6 digitálních pinů (3, 5, 6, 9, 10, 11) lze použít pro PWM⁶ výstup.

Programování desek probíhá skrze multiplatformní vývojové prostředí Arduino IDE. Desky jsou k počítači připojeny pomocí USB rozhraní, které softwarově simuluje sériový přenos přes linku RS-232. Arduino IDE vychází z jazyků *Processing* a *Wiring*, programovací jazyk Arduino je obdoba jazyka C nebo C++ doplněná o specifická pravidla a konstrukce. Základ každého programu tvoří funkce *setup()* a *loop()*:

- *setup()*: tato funkce je volána pouze jednou po začátku programu, nebo po resetu. Slouží pro inicializaci proměnných, nastavení I/O pinů na vstupní nebo výstupní, případně jiné nastavení knihoven apod.
- *loop()*: hlavní smyčka programu, která je volána opakovaně až do vypnutí desky, nebo resetu.

3.5 Senzory

K poskytování informací o reálném prostředí slouží senzory. V dnešní době se na trhu vyskytuje obrovské množství sensorů pro snímání různých fyzikálních veličin, událostí (vibrace, pohyb), přítomnosti chemických látek ve vzduchu apod. Následující výběr představuje malou část sensorů související s problematikou hydroponických a závlahových systémů.

- **Senzor půdní vlhkosti**

Senzor slouží ke zjištění množství vody obsažené v půdě. Nejčastěji se využívá tzv. odporová metoda, která funguje na principu průchodu elektrického proudu přes půdu pomocí dvou sond a měřením elektrického odporu. Vlhká půda je polovodivý materiál, jehož vodivost vzrůstá se zvětšováním vlhkosti.

- **Senzor pH**

Hodnota pH udává stupeň kyselosti nebo zásaditosti půdy v rozmezí 0 až 14 (pH = 7 neutrální prostředí, pH < 7 kyselé prostředí, pH > 7 prostředí zásadité). K měření se používá potenciometrie s využitím skleněné elektrody jako měrného členu. Metoda je zaměřena na měření elektrického potenciálu mezi měrnou a referenční elektrodou.

⁶Pulzně šířková modulace

- **Modul měření teploty, vlhkosti a barometrického tlaku BME280**
 Sensor od společnosti Bosch umožňuje měření teploty v rozsahu -40 až 85°C s přesností $0,01^{\circ}\text{C}$, vlhkosti vzduchu a tlaku v rozmezí $300 - 1100\text{hPa}$. Pro komunikaci poskytuje I²C a SPI rozhraní. Sensor se vyznačuje vysokou přesností měření.
- **Senzor teploty a vlhkosti vzduchu DHT-22**
 Sensor kombinující v sobě měření teploty a vlhkosti, který pracuje pouze po jednom vodiči (*1-wire*). Dovoluje měření vlhkosti v rozsahu 0% - 100% s přesností 2% a měření teploty od -40 do $+80^{\circ}\text{C}$ s přesností $0,5^{\circ}\text{C}$.



Obrázek 3.12: zleva senzor půdní vlhkosti, DHT-22 a HC-SR04 (zdroj: Ebay.com)

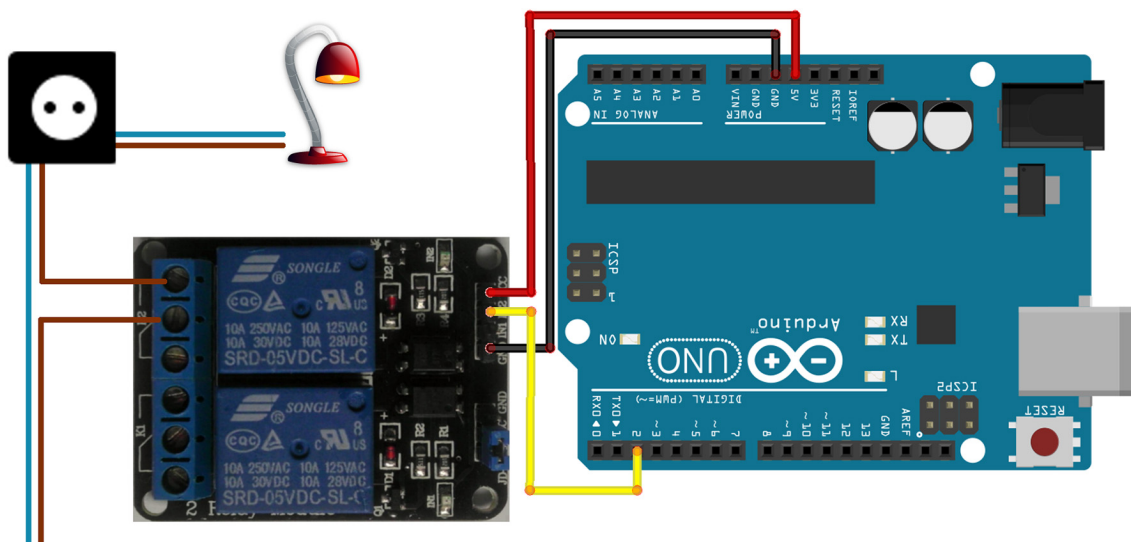
- **Senzor teploty a vlhkosti HTU21D**
 Levný, snadno ovladatelný a velmi přesný digitální vlhkoměr a teploměr. Komunikaci probíhá pomocí I²C sběrnice.
- **Digitální čidlo teploty Dallas DS18B20**
 Teplotní čidlo s vynikající přesností měření kolem $0,1^{\circ}\text{C}$. Dokáže měřit teploty v rozsahu -55 až $+125^{\circ}\text{C}$. Čidlo vrací přímou hodnotu teploty, není tedy nutný její přepočítání. Každé čidlo má svoji specifickou 64-bitovou adresu, díky čemuž je možné zapojit několik čidel na jeden kabel za sebe (podporuje „parasite power“). Komunikuje přes 1-vodičovou sběrnici (*1-wire*).
- **Fotorezistor**
 Fotorezistor je elektronická součástka, jejíž odpor se snižuje se zvyšující se intenzitou dopadajícího světla.
- **Senzor pro detekci vody**
 Pomocí tohoto senzoru je možné detekovat přítomnost vody například při nežádoucím úniku z elektrospotřebičů, haváriích na vodovodním potrubí, nebo při povodních. Jiným praktickým využitím je detekce deště. Sensor se skládá z plastové destičky s několika vodorovnými vodiči. Jakmile na destičku dopadne voda, některé z těchto vodičů se propojí a obvodem začne procházet elektrický proud.
- **Mechanický senzor výšky hladiny**
 Asi nejjednodušší a spolehlivý způsob měření výšky hladiny v nádobě. Využívá se plováku, který plave na hladině kapaliny a přes lanko je připojen k protizávaží vně nádoby. Pomocí převodníku a magnetických spínačů se potom určuje výška hladiny.
- **Ultrazvukový senzor vzdálenosti HC-SR04**
 Sensor funguje na principu odrazu od překážky. Vyšle se ultrazvukový pulz z vysílače,

který se odrazí od překážky a následně je detekován v přijímači. Změřením doby mezi odesláním a přijetím signálu se získá hodnota, kterou je potřeba vynásobit rychlostí zvuku a podělit dvěma, jelikož signál musí urazit cestu k překážce a zpět. Pracovní rozsah senzoru je 2 cm až 450 cm s přesností 3 mm.

3.6 Akční členy

Akční členy slouží k vykonávání akcí na základě vyhodnocení dat ze senzorů, nebo uživatelem.

Nejobecnějším akčním členem je **relé**. Relé je zařízení, které dovoluje ovládat mnohem větší napětí a proudy, než s kterým je schopný pracovat řídicí počítač/kontroler. Slouží například pro spínání čerpadel, žárovek, nebo klimatizace. Při vstupu logické hodnoty (nejčastěji 5V) je relé sepnuté, aby proud procházel nebo neprocházel v závislosti na zapojení. Pro použití s deskami Arduino existuje na trhu několik variant modulů s různými počty relé. Ukázkou takové desky včetně způsobu zapojení lze vidět na obrázku 3.13.



Obrázek 3.13: Relé modul připojený k Arduino Uno (zdroj: Browse-tutorials.com)

Kapitola 4

Specifikace systému pro automatizované pěstování rostlin

Tato kapitola se zabývá bližší specifikací systému pro automatizované pěstování rostlin. Je zde představen koncept celého systému jako bezdrátové sensorové sítě a její jednotlivé uzly. Také jsou vybrány základní technologie pro výsledný návrh a realizaci.

Na začátku byl zamítnut způsob hydroponického pěstování rostlin, ačkoliv se jedná o v dnešní době velmi populární řešení. Existuje zde totiž již velké množství konkurenčních produktů. Namísto toho bylo vybráno takové řešení, které pouze doplňuje současný stav ve většině domácností o moduly do půdy a nevyžaduje tak začít pěstovat rostliny od začátku.

4.1 Bezdrátová síť

Základem celého systému je bezdrátová síť sensorů a akčních prvků. Pro tuto práci bylo vybíráno z několika možností bezdrátové komunikace: Wi-Fi, Bluetooth Low Energy, ZigBee a rádiová komunikace pomocí NRF24L01. Níže jsou shrnuty jednotlivé varianty, jejich výhody a nevýhody:

- **Wi-Fi**

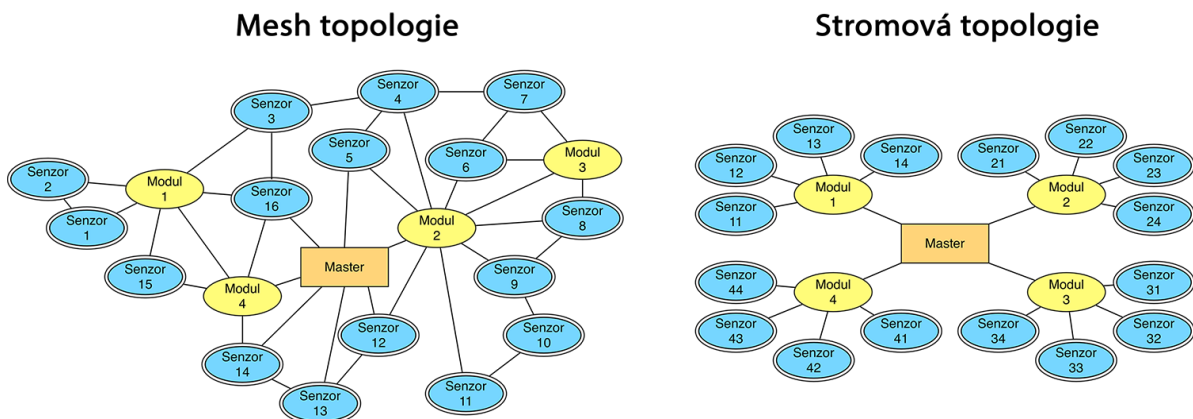
Bezdrátová komunikace pomocí technologie Wi-Fi a modulů ESP8266 je mezi Arduino uživateli velmi oblíbeným řešením. Umožňuje přímé připojení k domácí Wi-Fi síti a komunikaci do Cloudu například pomocí protokolu MQTT. Častým řešením je centrální prvek sítě v podobě jednodeskového počítače Raspberry Pi, na kterém běží MQTT broker a zpracovává tak požadavky ze sensorů. Broker může běžet také mimo lokální síť na vzdáleném serveru. Další alternativou je přímé připojení sensorů ke Cloudovým službám pomocí jejich protokolů. Dosah této technologie je určen dosahem lokální Wi-Fi sítě.

Výhody: přímé připojení k internetu, ESP8266 moduly lze rovnou programovat, nízká pořizovací cena

Nevýhody: vyšší energetická náročnost, nutnost složitější konfigurace každého modulu (uzlu sítě)

- **Bluetooth Low energy**

BLE oproti klasickému Bluetooth představuje na první pohled vhodné řešení. Pro sestavení klasické lokální bezdrátové sítě se ale příliš nepoužívá. Nejčastěji slouží pouze k jednoduchému propojení s mobilním zařízením, či jako podpůrný prostředek pro



Obrázek 4.1: Uvažované topologie bezdrátové sítě

konfiguraci bezdrátových modulů.

Výhody: jednoduchá komunikace, nízká energetická náročnosti, nízká pořizovací cena

Nevýhody: menší dosah, master/slave model s pouze 7 zařízeními typu slave

- **ZigBee**

Rádiová komunikace pomocí protokolu ZigBee a modulů XBee je asi nejlepší a nej-spolehlivější způsob bezdrátové komunikace z uvažovaných. Technologie umožňuje vytvořit topologii sítě hvězda, strom i mesh, definuje bezpečnost přenosu a má velký dosah (podle použitého typu antény). ZigBee je od počátku koncipováno pro použití v bezdrátových senzorových sítích. XBee moduly jsou ale oproti jiným alternativám mnohonásobně dražší.

Výhody: spolehlivá technologie, koncipována pro bezdrátové senzory, velký dosah

Nevýhody: vysoká cena XBee modulů

- **Rádiová komunikace pomocí NRF24L01**

NRF24L01 jsou často označovány jako levná alternativa k XBee modulům. Nativně není podporována topologie typu mesh jako u ZigBee, ovšem existují její softwarové alternativy. Díky své nízké ceně a oblíbenosti k nim bylo vytvořeno množství knihoven. Vyrábí se v několika variantách podle typu použité antény, čímž lze dosáhnout relativně velkého dosahu.

Výhody: nízká pořizovací cena, relativně velký dosah, nízká spotřeba

Nevýhody: neexistuje komunikační standard, bez uživatelských knihoven téměř nepoužitelné

Z výše uvedených možností byla vybrána varianta rádiové komunikace na 2,4GHz pomocí modulů NRF24L01. Původně byla uvažována technologie ZigBee s XBee moduly, ovšem toto řešení se ukázalo jako velmi cenově náročné vzhledem k rozsahu celého projektu. Moduly NRF24L01 jsou tak levnější alternativou. Také energetická náročnost je u těchto modulů mnohem nižší než například u Wi-Fi modulů ESP8266. Ty ovšem na druhou stranu nepotřebují ke svému chodu další mikroprocesor, jelikož jej přímo obsahují.

Pomocí zvolených modulů byla navržena centralizovaná bezdrátová síť s topologií typu mesh. Uvažována byla také topologie typu strom, kdy první úroveň by byla tvořena uzly napájenými ze sítě a k nim by byly připojeny bateriově napájené uzly. Správa takovéto sítě je ale náročnější na údržbu. V případě přesunu některého z koncových (bateriově napájených)

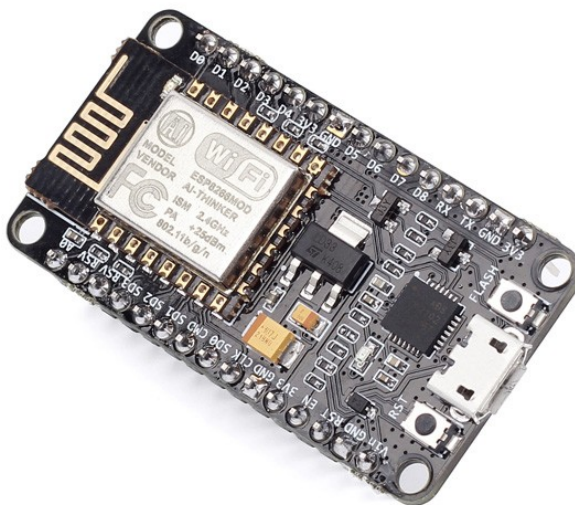
uzlů by bylo nutné změnit adresování tohoto uzlu v síti. Také výpadek uzlu na první úrovni by způsobil výpadek celé větve. Obě uvažované topologie jsou znázorněny na obrázku 4.1.

4.2 Gateway modul (koordinátor sítě)

Centrální prvek, v této práci označovaný jako **Master** nebo **Gateway modul**, je navržen jako koordinátor lokální sítě. Dále zajišťuje sběr dat ze sensorových uzlů, které budou představeny později, jejich zpracování a také jako brána do internetu. Celá síť je tak napojena skrze tento modul do Cloudu, aby byl umožněn vzdálený přístup pomocí mobilního zařízení. V Cloudu dochází také k vyhodnocování událostí na základě vstupních dat, které jsou odeslány zpět do master modulu a ten rozešle příkazy příslušným uzlům v lokální síti.

Požadavky na gateway modul:

1. Na jedné straně umožňuje komunikovat s lokální sítí rádiovou komunikací pomocí modulů NRF24L01.
2. Na straně druhé je připojen k internetu a komunikuje s Cloudovými službami.
3. Udržuje informace o lokální síti.
4. Slouží jako jistá forma DHCP serveru a spravuje připojené uzly.
5. Umožňuje párování a odpárování nového uzlu v lokální síti.



Obrázek 4.2: Modul NodeMCU (zdroj: arduinotech.cz)

Pro splnění podmínek se jevil jako nejvhodnější centrální prvek jednodeskový počítač Raspberry Pi 3. Ten obsahuje Wi-Fi modul, stejně jako ethernetové rozhraní, čímž je možné jej jednoduše připojit do internetu. Přes SPI rozhraní lze napojit NRF24L01 modul pro lokální komunikaci. Na Raspberry Pi existuje knihovna RF24Mesh¹ pro mesh topologii

¹Dostupné z: <https://github.com/nRF24/RF24Mesh>

sítě, čímž by byly splněny požadavky 3 a 4, jelikož knihovna oba tyto body řeší. Ovšem po pečlivém zvážení bylo použití Raspberry Pi zamítnuto a to ze dvou důvodů. Zprv se jedná o zbytečně výkonné zařízení, jehož možnosti by tak nebyly zdaleka využity naplno. Ale především cena Raspberry Pi 3 je relativně vysoká vzhledem ke svému účelu. Celý systém by se tak v nákladech značně prodražil a přitom by nebylo využito plného potenciálu.

Pro výsledné řešení byl navržen Wi-Fi modul NodeMCU. Ten obsahuje komunikační modul ESP8266 ve verzi ESP-12, 128kB operační paměti, 4MB uložení, USB rozhraní a GPIO piny. Programování probíhá v jazyce Lua, ovšem lze jej programovat také skrze rozhraní Arduino IDE stejně jako Arduino. Svým výkonem i množstvím periférií je tento modul vhodný jako řídicí mikroprocesor gateway modulu. Skrze SPI rozhraní lze připojit NRF24L01 modul a také je pro něj kompatibilní výše zmíněná knihovna RF24Mesh.

4.3 Půdní senzor

První ze dvou navržených bezdrátových modulů je půdní senzor (modul), dále označovaný pouze jako **Senzor**. Svojí podstatou i funkcí je velmi podobný zařízením Parrot Flower Power a Xiaomi Mi Plant představených v kapitole 2. Jedná se o bateriově napájený modul, postavený na mikroprocesoru od společnosti Atmel a komunikačním modulu NRF24L01. Pro měření byl navržen senzor půdní vlhkosti, fotorezistor pro zjištění množství dopadajícího světla a digitální čidlo teploty Dallas DS18B20.

Požadavky na půdní senzor:

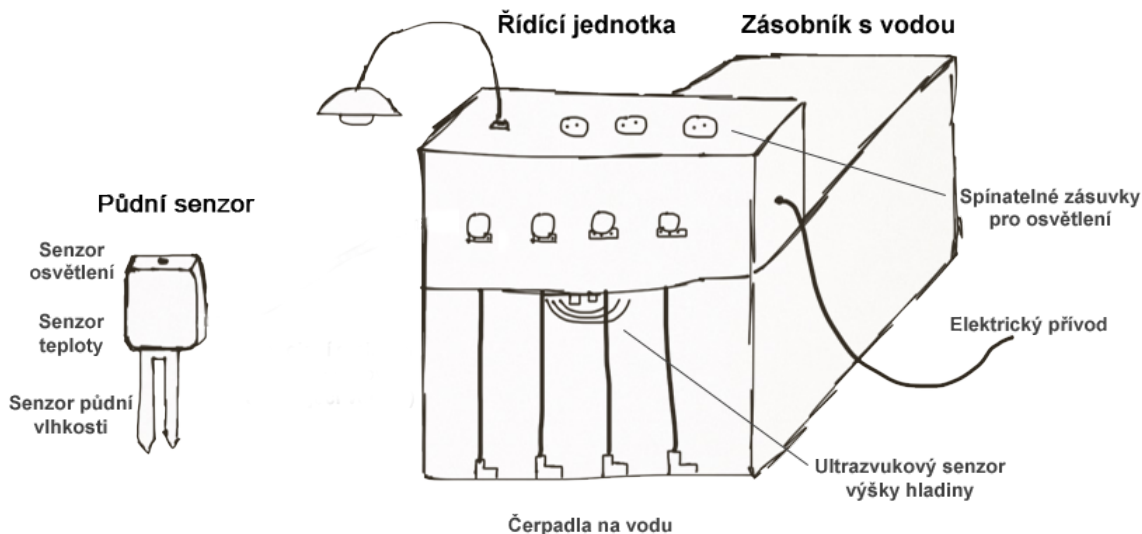
1. Velmi nízká spotřeba energie.
2. Napájení z 3V knoflíkové baterie CR2032.
3. Malé rozměry z důvodu umístění v květináčích.
4. Odolnost vůči vodě a vlhkosti.

4.4 Akční modul pro automatizované pěstování rostlin

Komplexnější **Modul** pro pěstování umožňuje automatické zalévání rostlin a umělé osvětlení na základě hodnot z půdního senzoru. Nákres navrženého systému lze vidět na obrázku 4.3. Jádrem je zásobník s vodou spolu s řídicí a komunikační jednotkou. Jako řídicí jednotka byl vybrán jednodeskový počítač Arduino. Jeden modul obsahuje 4 čerpadla a 4 spínatelné zásuvky na 230V. Čerpadla jsou vyvedena z modulu ven, kde je možné připojit rozvody vody do květináčů. Problém při zalévání může nastat v případě nedostatečného množství vody v zásobníku. Proto byl navržen ultrazvukový senzor HC-SR04, který dokáže určit výšku hladiny vody a v případě potřeby informovat uživatele.

Požadavky na akční modul:

1. Napájení z klasické 230V zásuvky.
2. Oddělení silnoproudé a slaboproudé části.
3. Silnoproudá část musí být izolována, aby nedošlo ke styku s vodou.
4. Kontrolní mechanismy aby čerpadla neběžela naprázdno bez vody.



Obrázek 4.3: Nákres systému pro automatizované pěstování rostlin

4.5 Cloud a mobilní iOS aplikace

System je skrze gateway modul připojen ke Cloudu, kde je realizována řídicí logika. Pro volbu Cloudového řešení bylo uvažováno několik služeb. V první řadě jsou to servery určené přímo pro internet věci, jako služba ThingSpeak, nebo AWS IoT od společnosti Amazon. První zmiňovaná je zcela zdarma, pouze s omezením na 3 mil. uložených záznamů za rok a limitem 15 sekund pro aktualizaci dat. Společnost nabízí také speciální programy pro studenty a domácí použití, které jsou u menších projektů zdarma. Na straně druhé byla uvažována služba Firebase od společnosti Google, která byla nakonec pro návrh vybrána z několika důvodů. Součástí této práce je také mobilní aplikace pro platformu iOS. Služba Firebase poskytuje celou ucelenou řadu nástrojů, které usnadňují tvorbu mobilních a webových aplikací. Výčet služeb byl již zmíněn v kapitole 2. Mobilní aplikace má v celém systému sloužit jako vizualizační a konfigurační nástroj pro uživatele. Neslouží tedy primárně pro řízení, ale pouze k nastavování parametrů pro automatizovaný chod.

Kapitola 5

Návrh a realizace systému

Tato kapitola se zabývá detailnějším návrhem systému podle představené specifikace v předchozí kapitole a jeho realizací. Je zde popsána jak realizace fyzických komponent, tak jejich programové řešení, řídicí logika systému a také mobilní aplikace.

5.1 Bezdrátová lokální síť

Bezdrátová komunikace celého systému je tvořena pomocí komunikačních modulů NRF24L01, jak bylo zmíněno v předchozí kapitole (4) specifikace. Moduly byly postupně testovány na prototypovacím nepájivém poli s jednodeskovými počítači Arduino Uno, Arduino Nano a Arduino Pro Mini. K testování byly použity komunikační moduly jak s anténou integrovanou přímo na desce, tak s externí anténou. Moduly NRF24L01 vyžadují napájecí napětí 3,3V, ale dokáží pracovat i s 5V logikou. Většina Arduino desek ovšem pracuje na 5V (kromě 3,3V verze Arduina Pro Mini), ale obsahuje 3,3V regulátor a tak je potřeba si dávat pozor na správné zapojení napájení. Vyšší napětí by způsobilo zničení celého modulu. Z důvodu velkých výkyvů odběru proudu při vysílání (změna z $1\mu A$ až na stovky μA) bylo použito doporučené zapojení elektrolytického kondenzátoru o kapacitě $10\mu F$ mezi napájení a zem. Připojení komunikačního modulu NRF24L01 k desce Arduino Uno přes SPI rozhraní je zobrazeno na obrázku 5.1. Tento náskres je vytvořen v prototypovacím nástroji Fritzing¹, který byl použit pro prvotní návrh zapojení všech desek plošných spojů v této práci. Nástroj umožňuje jak grafickou reprezentaci obvodu v podobě modelu (viz. obrázek), tak také zjednodušené schéma zapojení.

Ke komunikačním modulům existuje několik softwarových knihoven pro Arduino. Nej-používanější z nich jsou od uživatele Maniacbug² a uživatele TMRh20³. Knihovny od druhého zmíněného uživatele jsou aktuálnější a mezi uživateli více používané. Také nabízí již zmiňovanou knihovnu RF24Mesh pro vytvoření mesh topologie a proto bylo vybráno jejich použití. Konkrétně se jedná o následující knihovny:

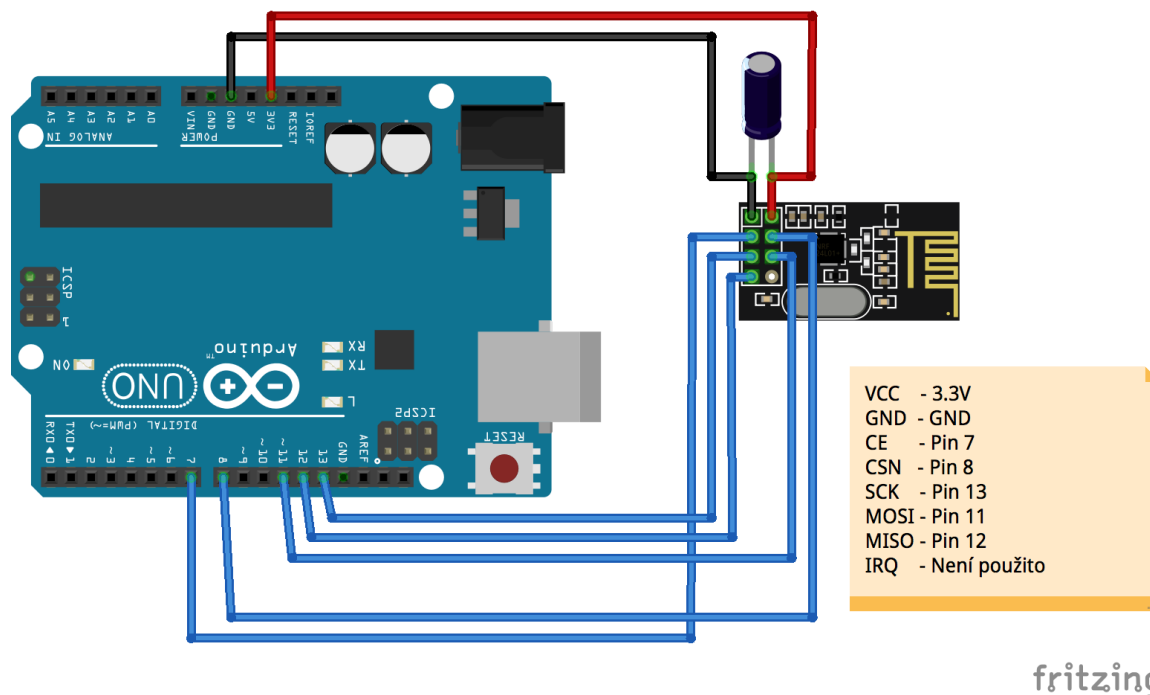
- **RF24**⁴ – základní knihovna pro práci s NRF24L01 moduly. Realizuje funkce pro komunikaci mezi moduly, jako odesílání zpráv, příjem zpráv, adresování, nastavení sítě a práci s komunikačními rourami (anglicky *pipes*). Těch může být současně otevřeno až 6, takže v jeden okamžik lze komunikovat až se šesti jinými zařízeními. Knihovna

¹Dostupné z: <http://fritzing.org>

²Dostupné z: <https://github.com/maniacbug/RF24>

³Dostupné z: <https://tmrh20.github.io>

⁴Více informací: <http://tmrh20.github.io/RF24/>



Obrázek 5.1: Zapojení komunikačního modulu NRF24L01 s Arduino Uno

dovoluje nastavení vysílací frekvence na jednu z 125 hodnot, představujících 125 kanálů mezi 2,4 a 2,525GHz. Dále lze nastavit vysílací výkon modulů na jednu ze čtyřech možností MIN, LOW, HIGH a MAX.

- **RF24Network**⁵ – knihovna realizuje síťovou vrstvu OSI modelu pomocí NRF24L01 modulů a využívá služeb předchozí knihovny RF24. Mezi základní funkce patří směrování a adresování v síti. Knihovna byla původně vyvinuta jako alternativa k protokolu ZigBee, který je použit u XBee modulů. Uzly v síti jsou adresovány pomocí 15-bitové adresy přiřazené administrátorem. Tato adresa přesně popisuje pozici uzlu ve stromové topologii sítě. Adresa centrálního uzlu je vždy 00, uzly k němu přímo připojené jsou adresovány jako 01, 02, 03, . . . , uzly na další úrovni 010, 011 atd. Ke komunikaci se potom používají právě tyto adresy a knihovna se stará o nalezení správné cesty mezi komunikujícími uzly.
- **RF24Mesh**⁶ – již několikrát zmiňovaná knihovna pro mesh topologii zajišťuje automatické adresování a dynamickou konfiguraci bezdrátové sítě. Uzly mají uživatelem definovanou unikátní adresu v rozsahu 1 – 255. Adresa číslo 0 je rezervována pro centrální uzel. Ten udržuje informace o celé síti a provádí adresování všech uzlů. Pro svůj běh využívá knihovna funkcí obou předchozích knihoven.

Všechny použité knihovny jsou určeny pro použití na platformě Arduino s mikroprocesory ATmega, na mikroprocesorech ATtiny, Linuxových zařízeních jako Raspberry Pi a existuje také jejich Python wrapper na Linux.

Pro centrální uzel, na kterém běží knihovna RF24Mesh a funguje jako jistá forma DHCP serveru, je předpokládáno použití jednodeskového počítače Raspberry Pi. Pro tuto práci

⁵Více informací: <http://tmrh20.github.io/RF24Network/>

⁶Více informací: <http://tmrh20.github.io/RF24Mesh/>

byl ale vybrán jako centrální uzel modul NodeMCU, čímž vzniká drobný problém v nekompatibilitě některých funkcí, které tak musely být pro tyto účely modifikovány. Jedná se hlavně o ukládání dat do lokálního souborového systému. Centrální uzel si udržuje tabulku o přiřazených adresách v lokální síti a tuto tabulku si ukládá do souborového systému, aby nedošlo k její ztrátě při restartu či vypnutí zařízení. V knihovně je ale pouze podporováno ukládání na Linuxových zařízeních. Proto byly funkce rozšířeny o ukládání do souborového systému SPIFFS (SPI Flash File System) na modulech NodeMCU. K těmto účelům byly rozšířeny funkce `loadDHCP()` a `saveDHCP()` v souboru `RF24Mesh.cpp` knihovny.

Párování uzlů v síti

Knihovna RF24Mesh používá pro uživatelské adresování uzlů unikátní adresu v rozmezí 1-255. Tyto adresy musí být přiřazeny ručně před zahájením komunikace s centrálním (master) uzlem. Přiřazení většinou probíhá při nahrání samotného kódu do mikroprocesoru, případně přes sériové rozhraní. Moduly v této práci ale mají mít schopnost určité modularnosti, díky čemuž je možné do sítě dynamicky přidávat, či odebírat uzly nové. Nelze tak jednoduše přiřadit jednotlivým uzlům adresy předem a předpokládat, že se dvě stejné adresy nepotkají v jedné síti, obzvláště když se jedná o tak malé rozmezí adres. Stejně tak není možná konfigurace přes sériové rozhraní, jelikož běžný uživatel k němu zpravidla nemá přístup. Proto vznikl následující systém párování:

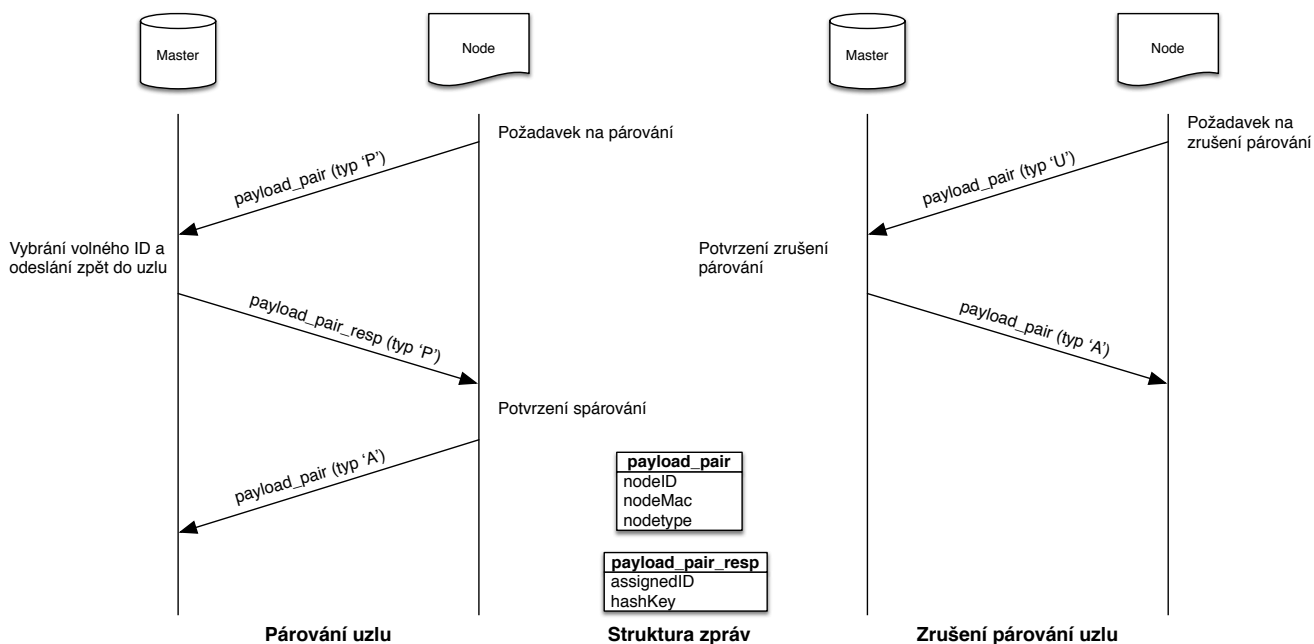
- Centrální, master, uzel má vždy adresu 0 a ostatní uzly jej tak dokáží jednoznačně identifikovat.
- Uzel, který není spárován do sítě má výchozí adresu 255.
- Po stisknutí fyzického párovacího tlačítka na master uzlu se aktivuje párovací mód, ve kterém modul očekává požadavek na spárování od uzlu s adresou 255. Každá zpráva v lokální síti je označena jednoznakovým typem. Požadavek na párování je označen typem 'P'.
- Po přijetí požadavku master uzel vyberou první volnou adresu z rozsahu 1-254, uloží si ji a odešle do párovaného uzlu zprávu s typem 'P' a nově přiřazenou adresou.
- Uzel si uloží přiřazenou adresu a odpoví zprávou s typem 'A', že informaci přijal. Následně provede odpojení od sítě, nastavení nové adresy a opětovné připojení k síti.
- Master uzel si potvrdí uložení přiřazené adresy do své lokální tabulky a ukončí párovací mód.

Centrální uzel si tak udržuje tabulku s přiřazenými adresami. Tato tabulka se ukládá vždy při změně do souborového systému, aby nedošlo ke ztrátě informací při restartu, či vypnutí zařízení. Kromě přiřazené adresy je každý uzel identifikován také unikátně vygenerovanou MAC adresou. Tu má uzel nastavenou již při nahrání programu a stejně jako v případě síťových karet se předpokládá, že se dvě stejné adresy nepotkají v jedné síti. Díky tomu lze identifikovat, zda modul byl již spárován do sítě, případně zda odpověď na párování přišla opravdu od původního odesílatele požadavku. Pokud by párovací požadavek dorazil z uzlu s jednou adresou, ale potvrzení od uzlu s adresou jinou, dojde ke smazání záznamu v tabulce a uzel tak není validně spárován. U jednotlivých uzlů se dále rozlišuje jejich typ. Typ 0 značí gateway modul (centrální prvek, master), typ 1 je akční modul (neuspává se) a typ 2 bateriově napájený senzorový modul (uspává se).

Kromě párování uzlů do sítě musí být možné uzel také odebrat. K tomu slouží navržený systém odpárování:

- Dlouhým stiskem fyzického tlačítka na odpárovávaném uzlu se odešle požadavek na zrušení párování (typ 'U') do uzlu master.
- Uzel master neustále kontroluje příchozí správy z lokální sítě a pokud obdrží zprávu označenou typem 'U' provede smazání záznamu ve své tabulce přiřazených adres. Následně odpoví zprávou typu 'A', aby informoval odesílatele o úspěšném odstranění.
- Uzel po obdržení potvrzení vymaže svoji původně přiřazenou adresu z nastavení, provede odpojení od sítě, změní adresu na výchozí hodnotu 255 a následně se do sítě připojí s touto adresou.

Grafické znázornění párování a odpárování, včetně struktury přenášených zpráv je zobrazeno na diagramech 5.2. Systém párování je velmi podobný procesu *Three-way Handshake* známého z ustavení TCP spojení. Ve struktuře zprávy `payload_pair_resp` si lze všimnout atributu `hashKey`, který obsahuje klíč pro šifrování dat v lokální síti. Ve výsledku ovšem tohoto klíče nebylo v práci využito, z důvodu menšího výkonu bateriově napájených modulů a složitosti šifrovacích algoritmů.



Obrázek 5.2: Párování a zrušení párování uzlů v síti

Knihovna pro párování a komunikaci

Výše uvedený způsob párování je realizován v knihovně `RF24Node`. Kromě párování zajišťuje knihovna i veškerou ostatní síťovou komunikaci. Využívá výše zmíněných knihoven třetích stran `RF24`, `RF24Network`, `RF24Mesh` a jejich instance očekává při inicializaci. Na začátku činnosti každého uzlu musí být volána metoda `beginNode`, respektive `beginCoordinator` v případě master uzlu. Tato metoda provede nastavení adresy, vysílačeho výkonu a spustí všechny využívané služby. V případě master uzlu se také inicializují

potřebné dynamické struktury. Pro ukládání adresy uzlů a šifrovacího klíče je využita nevolatilní paměť EEPROM. Zápis a čtení z této paměti je zajištěno knihovnou RF24Node.

Master uzel provádí skrze knihovnu také komunikaci s cloudovou službou Firebase, k čemuž využívá knihovnu *FirestoreArduino*⁷. Periodicky jsou kontrolovány nové data v cloudové databázi s intervalem 5s. Jelikož požadavek na událost může vzniknout pouze u akčních modulů, jsou kontrolovány nové data jen u nich. Všechny ostatní data se z lokální sítě do Cloudu pouze zapisují. Aby bylo možná detekovat, že se jedná o změnu některého ze stavů (zapnuto → vypnuto a naopak), udržuje si master uzel pomocnou tabulku s těmito stavy. Díky tomu se posílají akce lokální sítě do akčních modulů pouze v případě změny a ne každých 5s, jako je interval čtení z cloudu. Kromě čtení akcí na akčních modulech se provádí každých 15 minut kontrola, zda některý z uzlů (senzory, akční moduly) nebyl uživatelem smazán. Mazání totiž neprobíhá přímo, ale uzel je nejprve v databázi označen jako **Deleted** a teprve master uzel provede jeho fyzické odstranění. Tento způsob byl zvolen z jednoho hlavního důvodu: master uzel si před fyzickým smazáním z databáze smaže všechny informace o daném uzlu ze svých lokálních tabulek a provede tak odpárování ze sítě. Poslední funkcí, kterou master uzel skrze knihovnu realizuje, je přeposílání naměřených dat ze sensorů do cloudu. Půdní sensor a akční modul naopak skrze knihovnu provádí odesílání dat do masteru. Díky tomu, že sdílejí stejnou knihovnu, je zaručeno, že posílané struktury zpráv jsou vždy stejné, jelikož jsou definovány pouze na jednom místě. Stejná struktura zasílaných zpráv je pro správnou komunikaci kritická. Nejdříve je v hlavičce přijaté zprávy přečten její typ (jednoznakové označení), na základě něhož je rozpoznáno o jakou zprávu se jedná. K přečtení je potom nutné znát velikost přijaté zprávy, která odpovídá velikosti struktury s daty.

5.2 Datový model

Datový model celého systému byl navržen s ohledem na jeho fyzické součásti. Výsledná podoba je zobrazena na ER diagramu v příloze **B** a níže jsou popsány jeho jednotlivé části.

Tabulka Users

Ačkoliv je výsledný systém realizován pouze jednouuživatelsky, je zde navržena tabulka uživatelů. Kromě klasických atributů jako e-mail, jméno a příjmení, obsahuje také atributy `timeZone` s hodnotou časové zóny uživatele a `token` mobilního zařízení pro zasílání push notifikací. Ke každému uživateli může být přiřazen pouze jeden master modul.

Tabulka Masters

V této tabulce jsou uchovány informace o Gateway modulech. Primárním klíčem je MAC adresa uzlu. Atributy jsou naměřené hodnoty vlhkosti a teploty, časový údaj, kdy byla data naposledy aktualizována a dále obsahuje vazby na svoje spárované senzory a moduly.

Tabulka Modules

Tabulka `Modules` koresponduje s hardwarovou komponentou akčního modulu. Primárním klíčem je opět MAC adresa. Atribut `Paired` značí, zda je uzel aktuálně spárován v lokální síti a `Name` určuje výchozí název, který je vždy složen ze slova *Module* a číselného ID odpovídající přiřazené adrese v lokální síti. `CustomName` potom značí uživatelem definovaný

⁷Dostupné z: <https://github.com/firebase/firebase-arduino>

název. Atributy `Pump1-4` a `Relay1-4` jsou hodnoty typu `Bool`, které značí stav konkrétního čerpadla, respektive relé. `WaterLevel` udává výšku hladiny v zásobníku vody v cm a `WaterLevelStatus` potom vypočtený status výšky hladiny. Hodnota 0 představuje dobrý stav, 1 znamená upozornění na nízký stav vody a 2 velmi nízký stav vody nevhodný pro automatické zalévání. Tabulka má vždy jednu vazbu na master modul a maximálně 4 vazby na senzory.

Tabulka Sensors

Zde jsou uchovány informace z půdního senzoru. Primárním klíčem je MAC adresa uzlu. `Bool` hodnota `Battery` udává informaci o stavu baterie. Způsob zjišťování napětí na baterii bude popsán dále. Atributy `Light`, `Soil` a `Temperature` obsahují naměřené hodnoty ze senzorů. `AutoRelay` a `AutoPump` značí, zda se má provádět automatické spouštění osvětlení skrze relé a automatické zalévání. Oba atributy dávají smysl pouze v případě, že je daný senzor spárován s jedním ze slotů akčního modulu. Atribut `SoilStatus` nabývá hodnoty 0 v případě, že vlhkost půdy je dostatečná, 1 pokud se vlhkost začíná blížit suché půdě a 2 pro velmi suchou půdu. `TemperatureStatus` udává stav teploty a může mít hodnotu -2, -1, pokud je teplota příliš nízká, 0 při optimální teplotě a 1, 2 pokud je teplota naopak vysoká. Oba poslední atributy určující stav vlhkosti půdy a teploty jsou závislé na konkrétním nastavení pro zvolený druh rostliny, se kterou je senzor použit. Tabulka tedy obsahuje vazbu do tabulky `PlantDB`, kde jsou uloženy jednotlivé druhy rostlin a jejich specifické nastavení pro pěstování. Více o určování těchto stavů bude popsáno v části cloudové logiky systému 5.6.

Tabulka Data

V této tabulce se uchovávají historické naměřené hodnoty na sensorovém uzlu. Primárními klíči jsou MAC adresa a časové razítko kdy byla data naměřena. Atributy odpovídají třem měřeným hodnotám `Light`, `Soil` a `Temperature`. V této tabulce je také atribut `Timestamp` reprezentován opravdu jako UNIXový timestamp, tedy počet sekund od 1.1.1970. Jinde v datovém modelu je uložen pouze v textové podobě ve formátu "`YYYY-MM-dd HH:mm:ss`". UNIXová timestamp je zde uložen z důvodu snazšího filtrování dat. Jeho textová verze je zase lepší pro ukládání do mobilní aplikace.

Tabulka PlantDB

Poslední tabulkou je databáze jednotlivých druhů rostlin. Primárním klíčem je automaticky vygenerovaný klíč. Každá rostlina je určena názvem, svým jednoduchým textovým popisem a obrázkem. K rostlině je dále nastavena hodnota optimálního množství osvětlení `Light`, vlhkosti půdy `Soil` a minimální (`TempMin`) a maximální (`TempMax`) teplota pro pěstování.

5.3 Gateway modul

Gateway modul je stěžejní prvek celé bezdrátové sítě. Spravuje ji, zajišťuje směrování a komunikaci, je připojen k internetu a interaguje s cloudovými službami.

Hardware

Základem je Wi-Fi modul NodeMCU, vybraný v kapitole 4. Skrze SPI rozhraní je k němu připojen komunikační modul NRF24L01 ve verzi s externí anténou. Tato verze byla vybrána z důvodu většího dosahu, který je pro centrální modul žádoucí. Modul dále obsahuje resetovací tlačítko, tlačítko pro párování nových uzlů do sítě, 4 informační LED a jako bonus digitální senzor teploty a vlhkosti vzduchu HTU21D. NodeMCU obsahuje celkem 11 digitálních pinů, ale pro připojení všech periférií je jich potřeba 12 (5 pinů SPI, 1 pin pro tlačítko, 4 piny pro LED a 2 piny pro I²C rozhraní, skrze které je připojen senzor HTU21D). Z tohoto důvodu byl do obvodu přidán I²C expander PCF5874P. Ten umožňuje skrze I²C rozhraní rozšířit počet vstupně - výstupních pinů až na 64 (jeden expandér má 8 dodatečných pinů). Adresa expandéru se určuje 3 piny, čímž lze zvolit jednu z 8 adres. Pro tento obvod je adresa nastavena na 32 dekadicky, tedy všechny 3 adresovací piny jsou přivedeny na logickou nulu. Výsledné schéma obvodu, vytvořené v nástroji EAGLE, lze nalézt v příloze D.1.

Obvod je napájen z externího 9V zdroje, jehož napětí je skrze lineární regulátor sníženo na 5V a přivedeno na vstupní 5V pin modulu NodeMCU. Pro správnou funkčnost jsou před a za regulátorem zapojeny keramické kondenzátory o kapacitě 330nF a 100nF. Zajímavostí je napájení komunikačního modulu. Jelikož bylo plánováno použití nejvyššího vysílacího výkonu, byl do obvodu navržen také 3,3V regulátor jako externí zdroj napájení. Po zapojení ovšem vznikl problém, že modul takto napájený nebyl schopen komunikovat. Jakmile došlo ke krátkodobému napájení přímo z 3,3V pinu NodeMCU, komunikace se rozběhla a dále byla schopná fungovat pouze z externího zdroje napětí. Tento jev se nepodařilo vysvětlit a projevoval se jak v prototypu na nepájivém poli s různými napájecími zdroji, tak i ve výsledném obvodu, ačkoliv zem a všechny další piny byly správně propojeny. Z tohoto důvodu byl 3,3V regulátor odstraněn a použito napájení přímo z modulu NodeMCU. Během testování se neobjevily žádné anomálie při vysílání. 10μF kondenzátor, zapojený před vysílacím modulem, má dostatečnou kapacitu pro vykrytí i velkých špiček.

Navržený obvod byl postupně otestován na nepájivém poli a následně napájen na prototypovací desku plošných spojů. Deska byla zasazena do speciální plastové krabičky určené pro routery a jiná podobná zařízení. Ta svým tvarem naprosto splňuje parametry gateway modulu a obsahuje mimo jiné poloprůhlednou část pro 4 LED. Výslednou hardwarovou modobu gateway modulu lze vidět na obrázcích 5.3.

Software

Řídící program pro gateway modul je napsán v jazyce Arduino a využívá vlastní knihovny RF24Node. Jazyk Arduino byl zvolen namísto výchozího jazyka Lua hlavně z důvodu kompatibility knihovny. Lze tak použít stejnou knihovnu pro všechny moduly v síti a není třeba ji přepisovat do jiného jazyka. V knihovně je realizováno výše zmíněné párování/odpárování uzlů, komunikaci lokální síti a také komunikace s cloudovou službou Firebase.

Modul je skrze ESP8266 připojen do Wi-Fi sítě a internetu. K tomu je využito knihovny *ESP8266Wi-Fi*⁸. Knihovna předpokládá nastavení SSID názvu sítě a hesla pro připojení již v době překladu programu. To je ovšem z uživatelského hlediska velmi nepraktické. Proto byla použita knihovna *Wi-FiManager*⁹, která tento proces velmi usnadňuje. Jakmile ESP inicializuje svou činnost, pokusí se knihovna připojit k poslední uložené síti. Pokud

⁸Dostupné z: <https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266Wi-Fi>

⁹Dostupné z: <https://github.com/tzapu/Wi-FiManager>



Obrázek 5.3: Gateway modul

tento proces selže (jedná se o první spuštění, Wi-Fi síť není dostupná, poslední nastavení bylo smazáno) aktivuje se režim přístupového modu, ve kterém na modulu běží DNS a webový server. Uživatel se libovolným Wi-Fi kompatibilním zařízením připojí k tomuto přístupovému bodu. Po připojení se zobrazí konfigurační webová stránka, kde uživatel může vybrat některou z existujících Wi-Fi sítí v okolí, nebo může zadat přihlašovací údaje ručně. Poté se ESP pokusí připojit ke zvolené síti a pokud je úspěšné, provede se přepojení zpět na původní síť a konfigurace se ukončí. Uložené nastavení Wi-Fi sítě lze smazat dlouhým stiskem párovacího tlačítka.

Uživatel je o stavu modulu informován prostřednictvím 4 LED:

1. Zelená LED – stavová dioda

Svítlí – modul je připojen k Wi-Fi síti.

Bliká pomalu – modul není připojen k Wi-Fi síti, má uložené nastavení a snaží se připojit.

Bliká rychle – modul je ve stavu přístupového bodu. Nemá uložené nastavení Wi-Fi sítě, čeká na konfiguraci uživatelem.

2. Modrá LED – LAN

Informuje o komunikaci v rámci Wi-Fi sítě (komunikace s cloudovými službami). V případě, že je na modulu aktivován párovací mód, dioda bliká.

3. Modrá LED – PAN

Informuje o komunikaci v rámci lokální bezdrátové sítě pomocí NRF24L01 modulů.

4. Červená LED – chybová dioda

V případě, že nastane libovolná chyba, nejčastěji při lokální komunikaci, nebo komunikaci s cloudovými službami, rozsvítí se dioda po dobu 1s.

LED jsou ovládány skrze expandér PCF8574P, ke kterému vznikla jednoduchá knihovna *PCF8574Lib* usnadňující práci s ním. Data do expandéru jsou odesílána jako jedno binární

číslo, kde 1 na určité pozici představuje logickou hodnotu 1 daného pinu, což je i z programátorského hlediska nepraktické. Knihovna proto zajišťuje správnou tvorbu těchto binárních hodnot, které jsou následně odeslány do expandéru přes I²C rozhraní. Programátor tak na venek pracuje pouze s konkrétními indexy pinů, které chce adresovat. Kromě toho knihovna poskytuje funkce pro blikání na jednotlivých pinech, tedy periodickou změnu logické hodnoty.

Interakce s modulem probíhá pomocí jediného párovacího tlačítka. Jeho stav je opakovaně v programu kontrolován a rozlišují se dva druhy stisknutí. Krátkým stiskem se aktivuje párovací mód. V tomto módu modul očekává párovací zprávu od jiného uzlu v síti, jak bylo popsáno v části Bezdrátová lokální síť 5.1. LAN dioda bliká, čímž je informován uživatel. Opětným stisknutím se párovací mód manuálně zruší. Dlouhým stiskem tlačítka nad 3s se provede smazání nastavení o Wi-Fi síti a restart modulu. Druhé tlačítka na modulu slouží k ručnímu restartu NodeMCU. V programu se s ním tedy nijak neparacuje.

Kromě správy lokální bezdrátové sítě obsahuje modul také senzor teploty a vlhkosti HTU21D. Čtení dat probíhá přes I²C sběrnici, prostřednictvím knihovny *SparkFunHTU21D*¹⁰ každých 15 minut a data jsou okamžitě odesílána do cloudové databáze.

5.4 Půdní senzor

Půdnímu senzoru bylo v této práci věnováno asi nejvíce pozornosti. Jedná se o relativně malé zařízení, které je navíc bateriově napájeno.

Hardware

Jako řídicí mikrokontroler pro modul byla nejprve uvažována některá z menších desek Arduino (Arduino Nano, Arduino Pro Mini). I přesto, že jsou tyto desky relativně malé, stále nejsou dostatečně malé pro účely modulu. Navíc obsahují řadu podpůrných obvodů, diod apod., které mohou zbytečně odebírat elektrický proud. Proto bylo zvoleno použití pouze samotného mikroprocesoru a ne celé hotové desky. Na modulech Arduino jsou použity mikroprocesory Atmel, konkrétně ATmega328 (nejčastěji). Pro senzorový uzel byly v první fázi zvoleny mikroprocesory ATTiny84 od stejné společnosti, které jsou menší a méně výkonné než ATmega, ale lze je programovat stejným způsobem. ATTiny84 obsahuje 14 pinů, jež přesně pokrývají potřeby modulu. Po vytvoření schématu a otestování se ovšem ukázaly dva zásadní problémy. Mikroprocesor obsahuje pouze 8KB programové paměti, což je nedostatečné pro všechny použité knihovny a dále knihovna RF24Mesh není kompatibilní s touto architekturou. Proto bylo schéma změněno na použití stejných mikroprocesorů ATmega328 jako na deskách Arduino. Díky svojí rozšířenosti jsou paradoxně levnější než dříve použité ATTiny84, přitom poskytují mnohem více vstupně - výstupních pinů a také vyšší výkon.

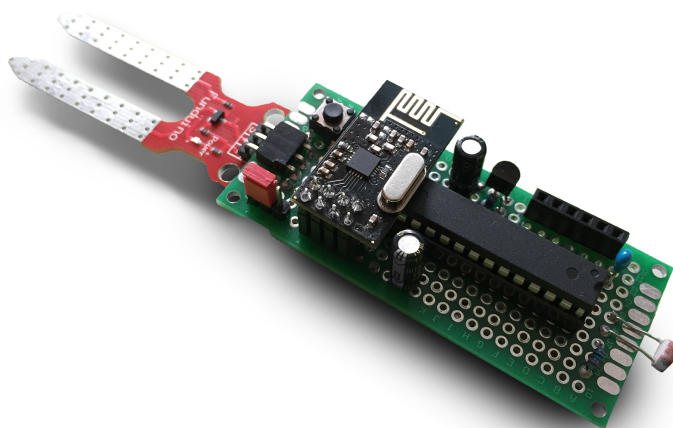
Dle prvotní specifikace modul obsahuje 3 druhy senzorů – senzor půdní vlhkosti, fotorezistor a teplotní čidlo Dallas DS18B20. Pro správnou funkci fotorezistoru je potřeba k němu připojit pull down odpor, jehož hodnota určuje rozlišovací schopnost fotorezistoru. Při vyšší hodnotě dokáže lépe rozpoznat změny při slabších světelných podmínkách, při nižší hodnotě potom při větší intenzitě světla. Postupně byla otestována rozpoznávací schopnost s odpory 10k Ω , 5k Ω a 1k Ω . Jako nejlepší kompromis byla zvolena právě prostřední hodnota 5k Ω , kdy senzor při plném denním světle vrací maximální hodnotu, ale na druhou stranu dokáže lépe detekovat postupné stmívání. V průběhu vývoje se použití fotorezistoru ukázalo jako méně

¹⁰Dostupné z: <https://learn.sparkfun.com/tutorials/htu21d-humidity-sensor-hookup-guide/htu21d-library-and-functions>

vhodné, jelikož takto získanou hodnotu nelze jednoduše převést na jednotku intenzity osvětlení Lux. Pro tyto účely by bylo vhodnější použití fotodiody, která má přesně definovanou svou charakteristiku. Dále je na modulu informační LED, párovací tlačítko a komunikační modul NRF24L01. Pro účely programování mikrokontrolerů je v obvodu vytvořeno FTDI rozhraní se 6 piny.

Důležitým parametrem celého modulu je bateriové napájení. Mikroprocesory ATmega poskytují režimy spánku, díky čemuž je možné modul úplně uspat po předem definovaný čas. Problémem je fakt, že čas lze nastavit maximálně na 8s, což je relativně krátká doba. Standardně se uspání na delší dobu řeší softwarově, kdy se po 8s obvod probudí, zkontroluje se vnitřní časovač (proměnná), zda již nepřekročil definovanou hodnotu a pokud ne, obvod se opět na 8s uspí. Pouze v případě, že časovač překročí nastavený čas, dojde k samotnému vykonání nějaké akce. Tento způsob je použitelný, ale stále se obvod probouzí velmi často, což má negativní dopad na baterii. Po prozkoumání možností byly vybrány zajímavé časovače Nano Timer od společnosti Texas Instruments. Ty jsou určeny přesně pro účely bateriově napájených senzorů a internet věcí. Dovolují skrze tranzistor odpojit celý obvod od napájení po uživatelem definovanou dobu a samy v klidovém režimu odebírají pouze 35nA. Z nabídky byl konkrétně vybrán model TPL5110¹¹. Ten umožňuje definovat časový interval v rozmezí 100ms až 7200s pomocí odporu (hodnota odporu určuje časový interval, viz. datasheet). Aby bylo možné napájení obvodu manuálně sepnout, je zde navrženo start tlačítko, které právě tuto funkci realizuje. Schéma obvodu lze nalézt v příloze D.2.

Prototyp senzoru



Obrázek 5.4: Prototyp půdního senzoru

Půdní senzor vznikl postupně v několika prototypech. Nejprve byla odladěna funkčnost s deskou Arduino Pro Mini na nepájivém poli, poté bylo Arduino zaměněno za mikroprocesor ATmega328p-PU. Na většině Arduino desek je použit externí 16MHz krystal, jako zdroj

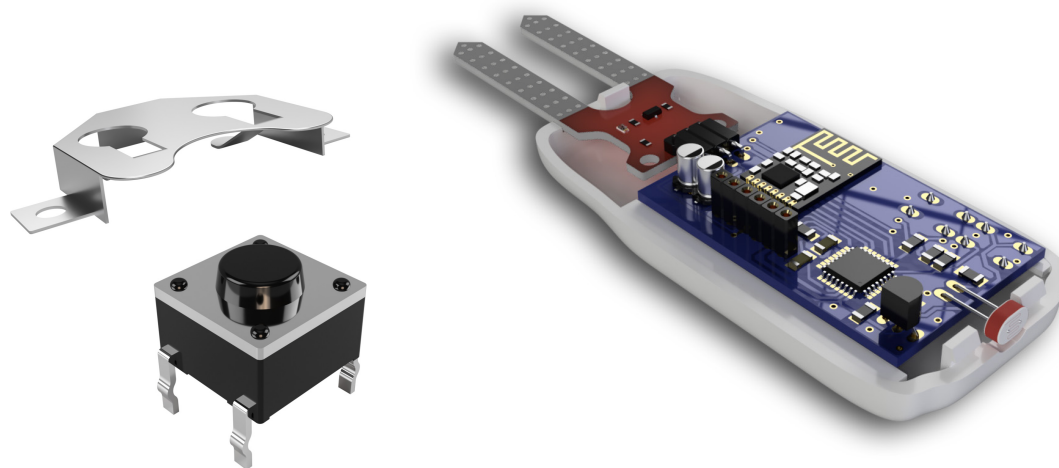
¹¹Datasheet dostupný z: <http://www.ti.com/lit/ds/symlink/tpl5110.pdf>

Jelikož se jedná o relativně malé zařízení a tvorba prototypu je z hlediska propojení všech komponent velice náročná, bylo rozhodnuto o vytvoření a vyrobení vlastní desky plošných spojů. K tomu bylo využito nástroje EAGLE a již hotového obvodového schématu. Řídící mikroprocesor byl zaměněn za SMD verzi s označení ATmega328p-AU, stejně tak komunikační modul byl zvolen v SMD verzi z důvodu velikosti a i všechny ostatní komponenty byly vybrány v jejich SMD verzi. K realizaci byla použita pouze dvouvrstvá deska, jelikož se nejedná o nikterak složitý obvod. Navíc cena výroby dvouvrstvé desky je podstatně nižší, než desek vícevrstevných. Rozmístění komponent bylo několikrát upravováno. Hlavním nedostatkem bylo špatné rozvržení kondenzátorů pro vykrytí změn v napájení. Ve výsledku je zde, kromě $10\mu\text{F}$ kondenzátoru pro komunikační modul, jeden $1\mu\text{F}$ elektrolytický kondenzátor pro celý obvod, dva menší 100nF keramické kondenzátory umístěné co nejbližší napájecím pinům mikroprocesoru a jeden 100nF keramický kondenzátor u fotorezistoru pro správné měření.

Výroba desek byla realizována v Číně službou Seeed Studio¹³. Tato služba nabízí možnost objednat výrobu online skrze jejich webové rozhraní, které mimo jiné obsahuje nástroj pro kontrolu dodaných podkladů. Samotná výroba trvá 5 pracovních dní a dodání dle zvoleného způsobu doručení od 1 do 22 dní. Vyrobene desky byly následně ručně osazeny, jak lze vidět na obrázcích 5.5a a 5.5b.

3D model půdního senzoru

V rámci práce vznikl také 3D model půdního senzoru a krabičky na něj. Na trhu nebyly nalezeny vhodné krabičky pro umístění a proto byla vymodelovaná krabička následně vytisknuta na 3D tiskárně.



(a) Držák baterie, mikrospínač

(b) Půdní senzor zasazený v krabičce

Obrázek 5.6: 3D model

K realizaci byl použit nástroj Fusion 360 od společnosti Autodesk. Postupně byly dle své výkresové dokumentace vymodelovány jednotlivé komponenty jako odpory, kondenzátory, mikroprocesor, komunikační modul, mikrospínače apod. Ukázkou mikrospínače a držáku

¹³Více informací: https://www.seeedstudio.com/fusion_pcb.html

baterie lze vidět na obrázku 5.6a. Následně byla vymodelována deska plošných spojů přesně podle gerber souborů sloužících jako podklady pro výrobu, čímž model odpovídá realitě. Dříve vymodelované komponenty byly posléze „osazeny“ na desku a virtuálně „zapájeny“. Díky tomu, že vše bylo vytvořeno podle přesných rozměrů, tak i výsledný model do sebe přesně zapadá. Sestavený model včetně krabičky je na obrázku 5.6b.

Následně byla kolem modelu vymodelována krabička. Ta vychází z volného tvaru kvádru, který byl postupnou deformací tvarován do výsledné podoby. Vnitřek tělesa byl vybrán a vytvořeny tak 1,8mm široké stěny. Poté došlo k rozdělení na horní a dolní část krabičky. Uvnitř jsou vymodelovány pomocné plošky pro uchycení desky plošných spojů a také uzavírací mechanismus v podobě háčků na spodní straně krabičky a otvorů v protikuse. Díky tomu lze krabičku jednoduše zavírat pouhým zacvaknutím obou dílů do sebe. 3D tisk byl realizován na FaVU VUT v Brně metodou FDM (*Fused deposition modeling*) z bílého ABS plastu. Model byl pro výrobu dodán ve formátu `.stl`. Aby nedošlo ke zlomení zacvakávacích háčků na krabičce, bylo nutné tisknout model na boku. Jednotlivé vrstvy materiálu tak nejsou nanášeny ve směru, kde je na háčky při otevírání vyvíjen největší tlak. Celková doba tisku na profesionální tiskárně činila asi 4h. Model a jeho výslednou fyzickou realizaci lze vidět na obrázcích 5.7a a 5.7b. Krabička bohužel nesplňuje požadavek odolnosti proti vodě a vlhkosti.



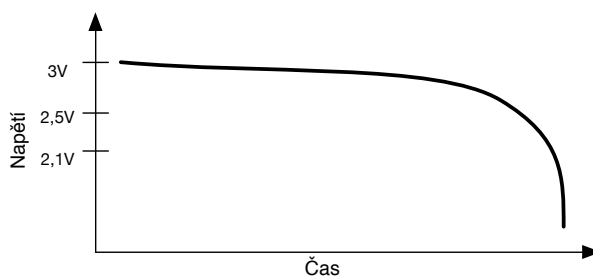
Obrázek 5.7: Půdní senzor

Software

Řídící program pro půdní senzor pracuje, stejně jako všechny ostatní uzly, z velké části s vlastní knihovnou RF24Node. Pro každý senzor je potřeba určit jeho specifickou MAC adresu na začátku programu.

Při návrhu hlavní funkce `loop()`, která běží opakovaně v nekonečné smyčce až do vypnutí, bylo nutné uvažovat jiným způsobem než u ostatních modulů. Senzor je vždy po přečtení sensorových hodnot a odeslání do gateway modulu uspán, čímž dojde k odpojení napájení a vypnutí mikroprocesoru. Po opětovném probuzení se provede vždy celý kód včetně nastavení adresy, připojení do sítě apod. v inicializační funkci `setup()` Arduino programu. V první iteraci hlavní smyčky dojde k přečtení a odeslání dat do gateway modulu, v pomocných proměnných se poznačí, že byla data opravdu odeslána a pokud došlo během odesílání k chybě, je tento proces opakován. Poté pokud není stisknuto tlačítko, nebo se modul necnachází v párovačím či odpárovačím módu dojde k uspání. To probíhá zasláním

signálu na DONE pin Nano timeru TPL5110. Před tím je korektně uvolněna přiřazená adresa v rámci mesh sítě. Maximální doba po kterou se modul pokouší data opakovaně odeslat je 6 sekund. V případě, že se během této doby odeslání nezdaří, dojde opět k uspání. Může se stát, že gateway modul není schopen komunikovat a v takovém případě je nežádoucí, aby sensorový uzel zůstal stále napájen. Stejně tak během inicializačního procesu (funkce `setup()`) je pomocí *watchdog časovače* (česky „hlídací pes“) kontrolována doba, po kterou se modul snaží připojit do sítě. Pokud dojde k vypršení časovače nastaveného na 8s, modul se uspí. Při testování bateriového napájení a probouzení modulů pomocí nano timeru bylo zjištěno, že v některých případech nejsou data korektně doručena do gateway modulu. Jako možná příčina byla zjištěna velikost posílané zprávy. Ta obsahuje MAC adresu uzlu, teplotu, množství dopadajícího světla, vlhkost půdy a informaci o stavu baterie. Jakmile došlo ke zmenšení zprávy, všechny data byla doručena vždy bez problému. Z tohoto důvodu bylo odesílání rozděleno mezi dvě samostatné zprávy. Knihovna *RF24Network*, která zajišťuje posílání dat v síti, provádí interní fragmentaci na menší segmenty, stejně jako v případě fragmentace IP datagramů v klasických počítačových sítích. V důsledku toho se pravděpodobně stávalo, že zpráva nebyla vždy doručena celá a tím pádem vůbec, naopak pokud je zpráva ručně rozdělena na dvě tak, aby k interní fragmentaci nedocházelo, bylo doručení vždy úspěšné. První zpráva obsahuje MAC adresu uzlu, teplotu a stav baterie. Druhá potom opět MAC adresu, aby bylo možné uzel detekovat, množství světla a vlhkost půdy. Pro čtení teploty ze senzoru DS18B20 je využito knihoven *OneWire* a *DallasTemperature*. Teplota, vrácená ze senzoru, je již přímo ve stupních Celsia na dvě desetinná místa. Pro účely zaslání do gateway modulu a dále do Cloudu je tato hodnota vynásobena 100 aby se předešlo problémům s čísly s plovoucí řádovou čárkou (float). Při testování byla zjištěna nekompatibilita datových typů na architektuře AVR (ATMega) a ESP8266 (NodeMCU), díky čemuž přijatá floatová hodnota na gateway modulu neodpovídala odeslané hodnotě ze senzoru. Získání informace ze senzoru vlhkosti a fotorezistoru je pouhé přečtení hodnoty z příslušného analogového pinu. Jedná se tedy číslo v rozsahu 0-1023, které samo o sobě musí být ještě správně reprezentováno. Více o interpretaci těchto hodnot bude uvedeno v části týkající se cloudové logiky systému.



Obrázek 5.8: Vybíjecí křivka baterie

Senzor dále zjišťuje stav baterie, aby bylo možné informovat uživatele o případné nutné výměně. Nejčastěji se využívá měření napětí na baterii, jelikož zjištění aktuální kapacity není úplně triviální úkol. Vybíjecí křivka standardních baterií má exponenciální průběh a je znázorněna na obrázku 5.8. Díky tomu nelze přesně určit aktuální stav baterie, ale pouze detekovat, že se blíží konec její životnosti. Napětí lze změřit jeho přivedením na analogový pin mikroprocesoru, kdy z přečtené hodnoty se trojčlenkou dopočítá hodnota napětí. Takto lze měřit baterie maximálně do 5V, což je největší přípustná hodnota napětí na pinech mikroprocesorů ATMega. Pro větší napětí je nutné použít napěťový delič (angl.

Voltage divider). Nejprve byla uvažována metoda měření pomocí analogového pinu, ale nakonec byl objeven způsob přímého měření čistě pomocí kódu. K tomu je využito interní napěťové reference 1,1V a následných výpočtů. Celý původní kód, včetně komentáře lze nalézt v Google Code¹⁴. Plně nabitá knoflíková baterie CR2032, použitá v senzoru, má napětí 3V. Jako krajní hranice byly zvoleny hodnoty 2,5V a 2,1V. Pokud napětí klesne pod první hranici je informován uživatel o nízkém stavu baterie, v případě, že poklesne i pod druhou hodnotu, senzor přestává komunikovat, aby nedocházelo k zasílání chybných zpráv.

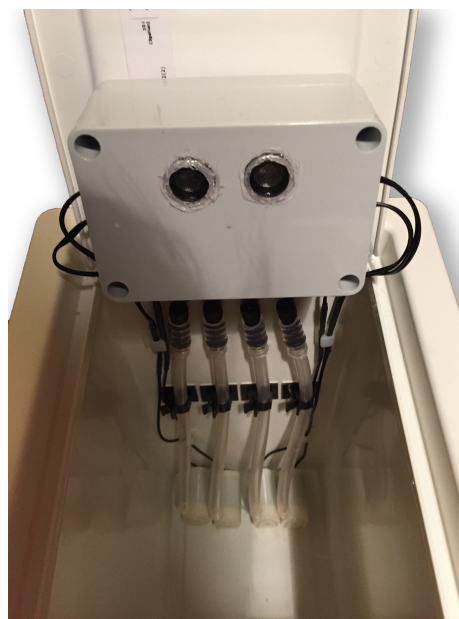
5.5 Akční modul

Poslední fyzickou komponentou systému je akční modul. Ten poskytuje rostlinám umělé osvětlení a automatizované zalévání.

Hardware



(a) Vnější pohled



(b) Vnitřní pohled

Obrázek 5.9: Zásobník vody

Realizace tohoto modulu je nejvíce náchylná na bezpečnost, jelikož obsahuje silnoproudé rozvody s 230V a zároveň zásobník s vodou. Původní vize plánovala realizaci jednoho boxu kombinujícího vše dohromady, tedy zásobník vodotěsně oddělený od řídicí části. Na trhu ovšem nebyl nalezen žádný odpovídající box splňující všechny parametry a jeho výroba na 3D tiskárně by byla příliš nákladná. Proto bylo rozhodnuto o oddělení obou celků do samostatných jednotek, propojených kabelem.

¹⁴<https://code.google.com/archive/p/tinkerit/wikis/SecretVoltmeter.wiki>

První část – **zásobník vody** byl vytvořen z odpadkového koše o objemu 10l a jeho vzhled je vidět na obrázku 5.9a. Je vybaven čtyřmi ponornými čerpadly vyvedenými z boxu ven, kde je připraveno připojení pro hadice do květináčů. Na otevíracím víčku koše je upevněna krabička s ultrazvukovým senzorem HC-SR04 a dvěma deskami s radiči motorů (*H-bridge*) (obrázek 5.9b). Každá z desek umožňuje řízení dvou motorů, v tomto případě čerpadel. Pro propojení s druhou částí modulu byl zvolen 8 pinový mikrofonní konektor z důvodu svojí bytelnosti a možnosti jej zašroubovat, takže nemůže dojít jednoduše k rozpojení.

Druhou část modulu tvoří **řídící box**, který lze vidět na obrázku 5.10a. K realizaci byla vybrána plastová průmyslová krabička s krytím IP65 (chráněno proti tryskající vodě). Na horní straně jsou umístěny čtyři zásuvky na 230V pro připojení libovolného umělého osvětlení. Každá ze zásuvek je spínána samostatně přes relé. Zásuvky byly vybrány s krytím IP54 (chráněno proti stříkající vodě) pro případ, kdy by box byl umístěn přeci jenom velmi blízko zásobníku, či jinému zdroji vody. Napájení je pomocí třípólového konektoru typu E, který je umístěn z boční strany boxu a obsahuje 10A trubičkovou pojistku. Dále se na boční straně nachází mikrofonní konektor pro propojení se zásobníkem vody, párovací a resetovací tlačítka, zelená stavová LED a modrá komunikační LED (obrázek 5.10b).



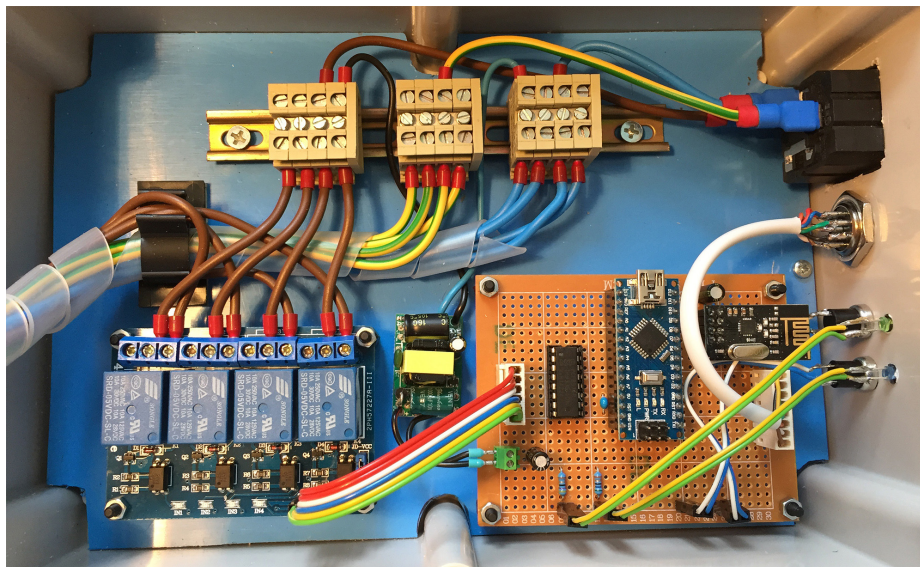
(a) Pohled z vrchu

(b) Pohled z boku

Obrázek 5.10: Řídící box akčního modulu

Řídící jednotka obsažená v boxu se skládá z desky Arduino Nano, komunikačního modulu NRF24L01 ve verzi s integrovanou anténou a I²C expandéru PCF8574P, jelikož Arduino neposkytuje dostatek digitálních pinů pro pokrytí všech potřeb. Schéma obvodu lze vidět v příloze D.3. Všechny komponenty byly napájeny na prototypovací desku plošných spojů, kde se nachází konektory pro připojení desky se 4 relé pomocí 6 pinového konektoru a 8 pinů pro připojení zalévacího boxu. Dále se zde nacházejí konektory se dvěma piny pro dvojici LED a tlačítka, které jsou v bočním panelu boxu. Pro napájení je použit 5V regulátor z 230V, přivedený na 5V pin Arduino. K pokrytí případných výkyvů v napájení je do obvodu přidán také 1 μ F elektrolytický a 100nF keramický kondenzátor. V příloženém schématu si lze také všimnout tranzistoru *Q1*, který byl do obvodu dodatečně přidán a slouží ke spínání napájení zásobníku vodu. Při testování bylo zjištěno, že po startu Arduino desky se na krátkou dobu asi 1-2s objevuje na výstupních pinech logická hodnota 1 a tím dochází ke krátkému sepnutí všech čerpadel. I přes přidání pull-down odporů k příslušným řídicím pinům čerpadel se tento jev stále vyskytoval. Proto byl zvolen zmíněný P-MOSFET

tranzistor jako spínač, který je skrze pull-up odpor připojen k napájení, čímž je zaručena logická hodnota 1 na hradle G v případě nedefinované hodnoty. K sepnutí tranzistoru potom dochází logickou hodnotou 0 po spuštění programu. Část vnitřního zapojení boxu je na obrázku 5.11



Obrázek 5.11: Vnitřní zapojení řídicího boxu

Software

Řídicí program v jazyce Arduino opět využívá vlastní knihovny RF24Node pro párování a bezdrátovou komunikaci a PCF8574lib pro práci s I²C expandérem. Na začátku programu je potřeba definovat unikátní MAC adresu daného modulu. Dalším parametrem pro nastavení je doba zalévání. Ta je fixně určena a její optimální hodnota byla zvolena testováním. Modul tak spouští čerpadla vždy po stejně dlouhou dobu, nehlédě na typu rostliny. Nelze totiž během zalévání kontinuálně měřit vlhkost půdy, zda již dosáhla požadované hranice, protože půdní senzory jsou usnuty a měří jen v určitých intervalech. V případě, že je vlhkost půdy po zalévání stále nedostatečná, projeví se to až při dalším probuzení senzoru. V hlavní smyčce programu `loop()` dochází každých 30s ke kontrole spojení v rámci lokální mesh sítě a případnému znovu připojení. Akční modul totiž běží nepřetržitě a v průběhu času může dojít k zneplatnění přiřazené adresy a modul tak není schopen komunikovat, ale hlavně přeposílat data v rámci mesh topologie.

Fyzická interakce s modulem je možná pomocí párovacího tlačítka. To rozlišuje, stejně jako půdní senzor, krátké a dlouhé stisknutí. Krátkým stiskem dochází k párování uzlu do sítě, dlouhým naopak odpárování. Stisknutí je detekováno přerušením a následným změřením doby trvání stisku. O tom, zda je modul připojen do mesh sítě, je uživatel informován zelenou LED, která se rozsvítí po úspěšném navázání spojení. Pokud probíhá bezdrátová komunikace, dojde po její dobu k identifikaci modrou LED. Informace o komunikaci je z knihovny RF24Node předávána pomocí tzv. „callbacku“, tedy částí kódu, která se provede voláním z jiného místa v kódu odkazem (referencí). Stejným způsobem předává knihovna do hlavního programu případné akce, které se mají na modulu vykonat. Akcemi je zapnutí/vypnutí některého z relé a zapnutí/vypnutí čerpadel. Jak již bylo řečeno, zalévání

probíhá po definované fixní dobu. Jakmile dojde k zapnutí čerpadla, poznačí se do proměnné aktuální čas a také, že čerpadlo běží. Hlavní smyčka programu neustále kontroluje, zda je některé ze čtyř čerpadel sepnuté a jejich případnou dobu běhu. V případě, že je přebročena, dojde k vypnutí čerpadla, resetování pomocných proměnných, odeslání informace o vypnutí do gateway modulu a dále do Cloudu a ke změření nové výšky hladiny. Gateway modul také resetuje atribut `soilStatus` u aktuálně zalité rostliny (senzoru). Zalévání se vůbec nespustí v případě, že v zásobníku není dostatečné množství vody. Aby se stavy relé a čerpadel po spuštění správně synchronizovaly s gateway modulem, je ihned po zahájení činnosti mikrokontroleru odeslána speciální zpráva, která informuje gateway modul, aby si uvedl všechny stavy do výchozí hodnoty vypnuto. Pokud je v cloudové databázi některé ze zařízení sepnuté, dojde v gateway modulu k detekování změny stavů a do akčního modulu se odešle žádost na sepnutí.

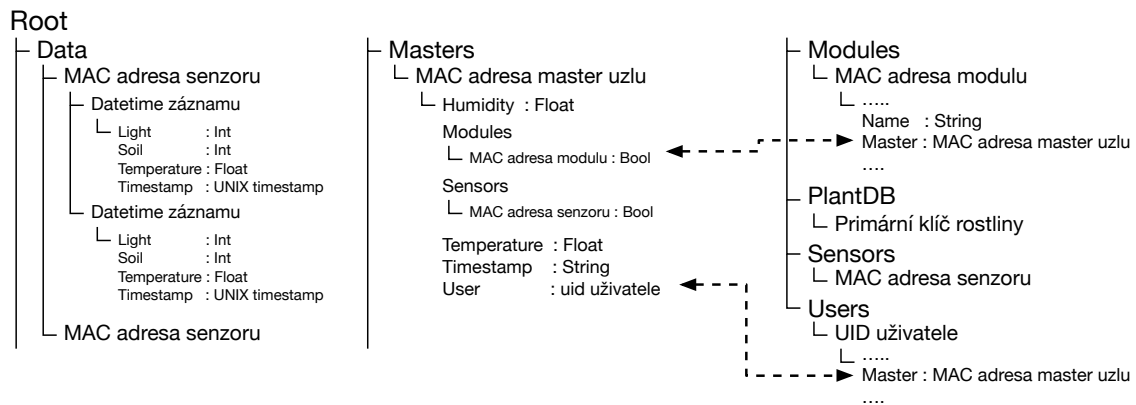
Měření výšky hladiny pomocí ultrazvukového senzoru je založeno na principu odrazu zvuku. Senzor obsahuje dva piny, jeden pro zaslání signálu (`trigger`) a druhý pro detekování jeho přijetí (`echo`). Na trigger pin se provede zápis logické hodnoty 1 (HIGH), čímž se vyšle pulz a následně pomocí funkce `pulseIn()` dojde ke změření doby přijetí pulzu na přijímači – pinu echo. Takto získaná doba v mikrosekundách musí být dále vynásobena rychlostí zvuku (uvažuje se rychlost $346,3 \frac{m}{s}$ v suchého vzduchu při teplotě $25^{\circ}C$) a podělena 2, jelikož pulz urazí vzdálenost od senzoru k překážce, kde se odrazí a zase zpět. Tímto způsobem je vypočtena konstanta 0,017315, kterou je změřená doba vynásobena a tím získána vzdálenost v cm.

5.6 Firebase a logika systému

V kapitole specifikace (4) byl jako Cloudová služba vybrán Firebase od společnosti Google. Ze všech nástrojů, které služba poskytuje, bylo využito především autentifikace uživatelů, realtime databáze, uložení (*Storage*), cloudových funkcí (*Functions*) pro logiku aplikace a notifikací (*Notifications*).

Ačkoliv je celý systém jednovýživatelový, je potřeba i tohoto jednoho uživatele přihlásit. Realtime databáze definuje bezpečnostní pravidla, kdy k datům není povolen přístup neautorizovaným uživatelům (lze změnit, ale z důvodu bezpečnosti se to nedoporučuje). Navíc do budoucna se plánuje celý systém zpřístupnit různým uživatelům a tak na to bylo již při návrhu myšleno. Data do databáze jsou ukládána podle datového modelu, představeného dříve. Je zde ovšem drobná odlišnost oproti klasickým relačním databázím. Data jsou uložena v jedné velké JSON struktuře, kde každá „tabulka“ tak představuje jednu větev struktury. V ní se potom nachází pole primárních klíčů jednotlivých záznamů. Vazby mezi „tabulkami“ jsou realizovány pouhou referencí název – hodnota. Na programátorovi tak je zajistit tyto vazby ručně. Problém může nastat v případě oboustranných vazeb mezi dvěma tabulkami. Je nutné zaručit správnost těchto vazeb u obou, vzájemně na sebe referujících objektů. Pro tyto případy Firebase umožňuje definovat kontrolní pravidla. Pomocí nich lze také ověřit správnost datových typů jednotlivých atributů, jelikož JSON umožňuje uložit libovolnou hodnotu. Těchto pravidel, zde ale nebylo využito a za správnost všech vazeb a datových typů zodpovídá mobilní aplikace, případně gateway modul, který data do cloudu ukládá. Ukázkou strukturalizace dat v realtime databázi lze vidět na obrázku 5.12. Šípkami jsou naznačeny atributy, které představují vazby mezi „tabulkami“. V rámci zjednodušení nejsou všechny vazby oboustranné, pouze ty, kde jsou opravdu potřeba.

Logika celého systému je realizována pomocí Firebase Functions. Všechny funkce jsou vykonány při změně určitého atributu v databázi (přidání/smazání/změna). Tato událost



Obrázek 5.12: Ukázka JSON struktury databáze

je označována anglickým názvem *Trigger*, česky spoušť nebo spouštěč. Následuje stručný popis jednotlivých funkcí:

- **temperatureStatus()**

Trigger: změna atributu **Temperature** v tabulce **Sensors**.

Akce: funkce nastavuje teplotní status (atribut **temperatureStatus**) pro daný senzor.

Podmínky: senzor musí být spárován s konkrétní rostlinou z tabulky **PlantDB**. Odtud se zjistí optimální teploty pro pěstování (atributy **TempMin** a **TempMax**), které jsou následně porovnány s naměřenou teplotou. Status -2 značí velmi nízkou teplotu pod hranicí **TempMin**, -1 nízkou teplotu v rozmezí **TempMin** a **TempMin + 3°C**, status 0 jsou optimální teplotní podmínky, 1 značí vysokou teplotu v rozmezí **TempMax - 3°C** a **TempMax** a nakonec status 2 představuje velmi vysokou teplotu nad **TempMax**.

- **soilStatus()**

Trigger: změna atributu **Soil** v tabulce **Sensors**.

Akce: funkce nastavuje status určující vlhkost půdy (atribut **soilStatus**).

Podmínky: senzor musí být opět spárován s konkrétní rostlinou, odkud je zjištěna optimální hodnota vlhkosti (**Soil**). Ta představuje procentuální vlhkost optimální půdy. Jelikož ze senzoru je přečtena pouze číselná hodnota v rozmezí 0-1023 bylo potřeba ji nějak interpretovat. Testováním se zjistilo, že hodnota 0 je vrácena pokud senzor není v půdě, hodnota menší než 300 značí suchou půdu, 300 - 700 optimálně vlhkou půdu a více než 700 půdu mokrou. Optimální hodnota **Soil** z tabulky **PlantDB** je potom porovnána s aktuální hodnotou vlhkosti půdy ze senzoru. Pokud je nižší než optimální hodnota je nastaven status na 2, tedy půda velmi suchá, v případě kdy aktuální vlhkost půdy je v rozmezí **Soil** a **Soil + 5%** jedná se o suchou půdu (status 1) a status 0 značí optimální vlhkost půdy. Př. Optimální hodnota pro danou rostlinu je 30%. Jako velmi suchá půda bude status vyhodnocen v případě, kdy ze senzoru vlhkosti bude přečtena hodnota menší než 300.

- **waterLevelStatus()**

Trigger: změna atributu **WaterLevel** v tabulce **Modules**.

Akce: funkce nastavuje status určující stav vody v zásobníku (atribut **waterLevelStatus**).

Podmínky: měřením bylo zjištěno, že výška prázdného zásobníku s vodou je asi 22cm \pm 1cm, jako nejistota měření v důsledku možných nepřesností způsobených přítomností čerpadel pod ultrazvukovým senzorem. Na základě toho byla určena minimální

vzdálenost mezi hladinou vody a senzorem na 20cm. Pokud je ultrazvukovým senzorem přečtena větší vzdálenost je status nastaven na hodnotu 2, tedy velmi nízký stav vody. V případě, že je vzdálenost 15 až 20cm je status roven 1, jako upozornění na nízký stav vody. Vzdálenost menší než 15cm představuje dostatečné množství vody v zásobníku a tedy status 0.

- **pumpStatus()**

Trigger: změna atributu **SoilStatus** nebo **Timestamp** v tabulce **Sensors**.

Akce: rozhodnutí o sepnutí čerpadla.

Podmínky: Aby bylo možné automaticky sepnout čerpadlo, musí být daný senzor spárován s akčním modulem a také musí být povoleno automatické zalévání (atribut **autoPump**). Pokud jsou obě podmínky splněny, zkontroluje se atribut **SoilStatus** senzoru, zda je roven hodnotě 2 a množství vody v zásobníku (atribut **WaterLevelStatus** roven 0 či 1). V případě, že jsou všechny podmínky splněny, dojde k sepnutí příslušného čerpadla.

- **relayStatus()**

Trigger: změna atributu **Timestamp** v tabulce **Sensors**.

Akce: rozhodnutí o sepnutí relé.

Podmínky: sepnutí relé vychází z informací o aktuálním množství světla ze senzoru, optimálním osvětlením pro zvolenou rostlinu a současné denní době. Senzor tedy musí být v první řadě spárován s rostlinou a akčním modulem a také musí být povoleno automatické osvětlení (atribut **autoRelay**). Aktuální hodnota osvětlení ze senzoru je před zpracováním váhovaně průměrována s předchozí hodnotou. Nová hodnota má dvakrát větší váhu než hodnota předchozí. Jelikož měření probíhá pouze jednou za čas, může se stát, že aktuálně přečtené množství světla neodpovídá příliš realitě. Například došlo ke krátkodobému zatažení, nebo naopak na senzor bylo v době měření více zasvíceno. Samotné rozhodnutí, zda osvětlení sepnout či vypnout, není úplně triviální úkol. Intuitivně by sepnutí probíhalo v případě, že množství světla poklesne pod předem definovanou mez. Pokud by se ale vycházelo pouze z informace o množství osvětlení není jasné, kdy je na senzor svíceno uměle nebo přírodně a proto je potřeba vzít v úvahu i tuto informaci. Dále ovšem nastává problém, kdy umělé osvětlení vypnout. Nelze experimentálně po nějakém čase zhasnout a čekat do dalšího probuzení senzoru, jestli je množství světla stále pod definovanou hranicí. A jak do celého systému zakomponovat specifický parametr pro konkrétní druh rostliny? Z těchto problémů byla vytvořena rozhodovací funkce, která pracuje následovně: nejprve je určena aktuální hodina (od 45. minuty je zaokrouhlena nahoru, např. 10:45 = 11h). Parametr pro konkrétní rostlinu určuje dobu, po kterou se má svítit, přičemž maximálně to bude do 6h ráno. Pokud je parametr nastaven například na 30, znamená to, že se bude svítit do $6h - ((100-30)/10)h = 23h$. V době 6h - 10h ráno je umělé osvětlení vždy zhasnuto. Takže i v případě, kdy parametr osvětlení je nastaven na 100%, bude rostlina na 4h denně bez umělého osvětlení. V ostatních časech je o sepnutí rozhodnuto na základě aktuální hodnoty osvětlení, která je porovnávána s předem definovanou hranicí. Jestli je umělé osvětlení již zapnuté, je naměřená hodnota větší než stanovená hranice, ale v tomto případě není žádoucí jej vypnout. Proto se akorát zkontroluje aktuální hodina a pokud je v rozmezí, po které má mít rostlina dostatek osvětlení, k vypnutí nedojde. V důsledku to znamená, že v případě kdy umělé osvětlení na základě nízkého množství světla sepne například už ve 12h, zůstane sepnuté až do maximální hodiny

určené parametrem osvětlení, nehledě na to, zda jsou po zbytek dne přírodní světelné podmínky dobré či nikoliv.

Firestore funkcemi je také realizováno ukládání historických naměřených hodnot. Spouštěčem funkce je změna atributu `Timestamp` v tabulce `Sensors`. Při změně se přečte nová hodnota atributů `Light`, `Soil` a `Temperature`, vypočítá se aktuální časové razítko v podobě UNIXového timestampu (počet sekund od 1.1.1970) a vše se uloží do tabulky `Data` pod příslušným primárním klíčem (MAC adresa) senzoru. Další pomocná funkce `insertPlant()` slouží pro vkládání rostlin do tabulky `PlantDB`. Jedná se o HTTP funkci, která se provede po zavolání příslušného HTTP příkazu. Jako parametr je předána JSON struktura, jež má být do tabulky uložena. Služba Firestore po nahrání takovéto funkce do Cloudu automaticky vygeneruje url adresu, na kterou se provádí daný HTTP požadavek. Pro inicializaci databáze rostlin byl vytvořen skript `initPlantDB.sh` spustitelný v prostředí Bash. Jeho zavoláním dojde k uložení několika výchozích druhů rostlin do tabulky `PlantDB` výše zmíněnou HTTP funkcí.

Službou Firestore Notifications spolu s Firestore functions je realizováno také **zasílání push notifikací** uživateli. Pro zasílání notifikací na mobilní platformu iOS je potřeba mít aplikaci zaregistrovanou ve svém Apple developerském účtu a následně vygenerovány Apple Push Notification service (APNs) SSL certifikáty. Celý postup zprovoznění Firestore notifiací na platformě iOS lze nalézt v dokumentaci služby Firestore¹⁵. Aby bylo možné zasílat notifikace na jedno konkrétní zařízení, musí být znám jeho `Token`. Ten je zjišťován po přihlášení uživatele do mobilní aplikace a následně uložen v databázi. Pokud dojde k odhlášení, je i tento token smazán, aby nedocházelo k odesílání notifikací na původní zařízení. Uživatel je pomocí notifikací informován o nízkém stavu vlhkosti půdy (pouze v případě kdy senzor není spárován s akčním modulem, nebo není povoleno automatické zalévání), o špatných teplotních podmínkách a o nízkém stavu vody v zásobníku.

5.7 iOS mobilní aplikace

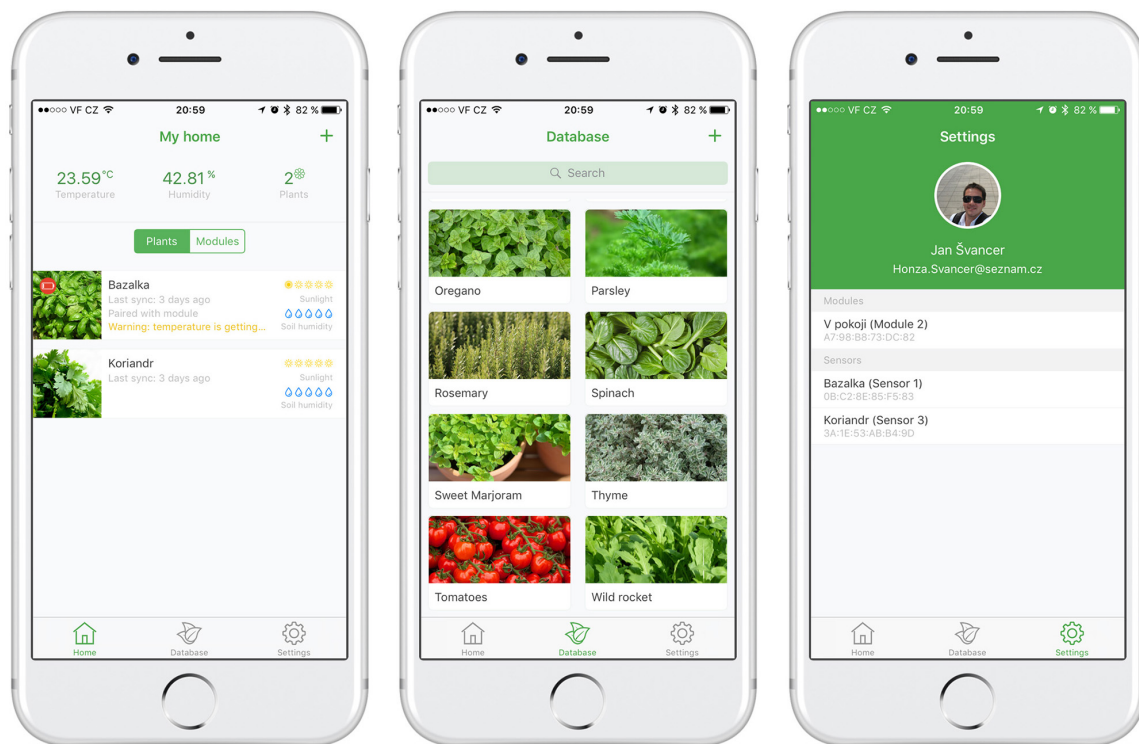
Poslední realizační částí je mobilní aplikace pro platformu iOS. Aplikace slouží jako vizualizační nástroj celého systému a také k nastavení jednotlivých parametrů.

Struktura aplikace a uživatelské rozhraní

Aplikace je rozdělena na 3 hlavní části. První z nich nazvána **My home** představuje přehled o celém systému. Jsou zde zobrazeny všechny rostliny (fyzické senzory spárované s rostlinou), akční moduly a jejich detaily. Druhá část **Database** je databází druhů rostlin a reflektuje databázovou tabulku `PlantDB`. Poslední částí je **Settings**, tedy nastavení. Zde jsou informace o přihlášeném uživateli a také všechny fyzické jednotky (senzory a moduly) spárované v síti, které je možné ze sítě smazat. Jeden uživatelský účet je vždy spjat s jedním master modulem. Samostatnou sekcí, stojící mimo hlavní aplikaci, je přidávání vlastních rostlin, neboli spárování fyzického senzoru s druhem rostliny z databáze a případným akčním modulem. Výslednou realizovanou podobu úvodních obrazovek jednotlivých částí aplikace lze vidět na obrázku 5.13.

Uživatelské rozhraní úvodní obrazovky aplikace odpovídá fyzickým komponentám systému. V horní části bylo navrženo zobrazení aktuální teploty a vlhkosti vzduchu z gateway modulu a dále přepínač mezi zobrazením všech senzorů spárovaných s rostlinami a všemi

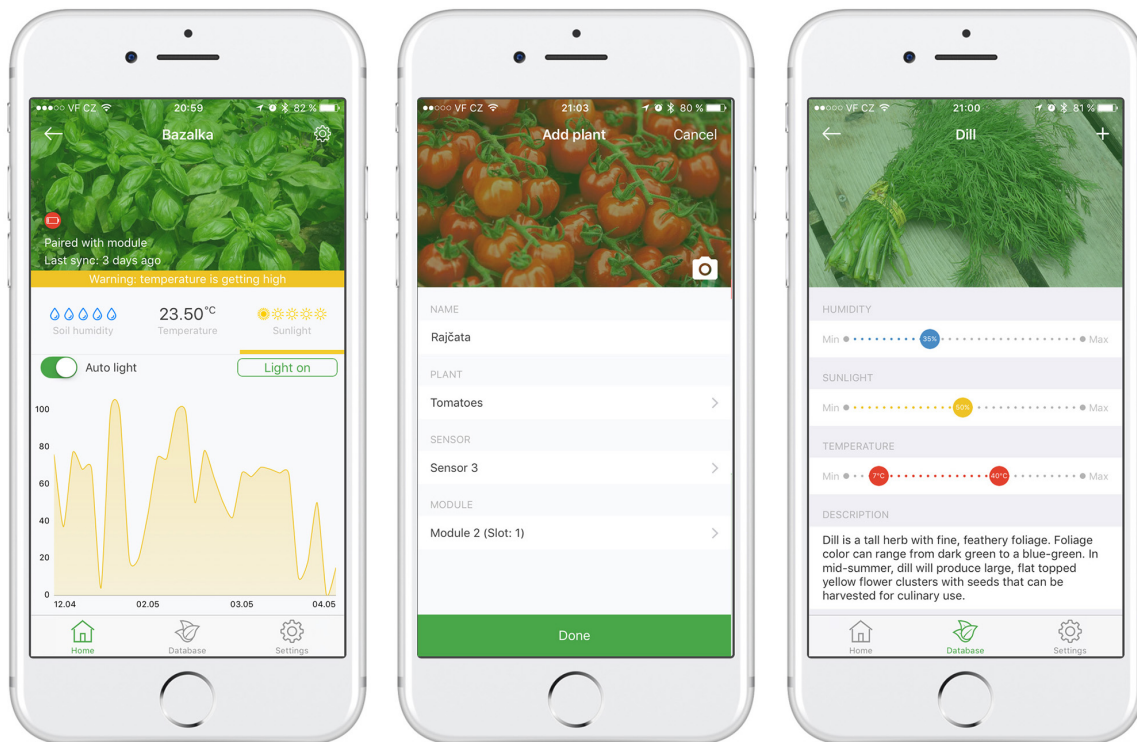
¹⁵Dostupné z: <https://firebase.google.com/docs/cloud-messaging/ios/client>



Obrázek 5.13: Hlavní sekce aplikace

moduly. Ty jsou zobrazeny v tabulce a obsahují nejpodstatnější informace, aby byly uživateli přístupné hned na první pohled. Řádek se senzorem zobrazuje jeho název, obrázek, čas od poslední synchronizace dat (kdy byla data ze senzoru naposledy přečtena), informaci zda je senzor spárován s akčním modulem, grafické znázornění nízkého stavu baterie a případné textové upozornění (nízká teplota, suchá půda apod.). Přímo v přehledu je také grafické znázornění aktuální vlhkosti půdy a množství dopadající světla. Rozkliknutím řádku tabulky se zobrazí detail. Zde jsou v podstatě stejné informace jako v přehledu, doplněné o aktuální teplotu a grafy naměřených hodnot za posledních 7 dní. Pokud je senzor spárován s akčním modulem obsahuje obrazovka přepínače pro automatické zalévání a automatické osvětlení a dále tlačítka pro manuální zálivku a manuální rozsvícení/zhasnutí osvětlení. V pravém horním rohu je navržena ikonka pro nastavení senzoru. Nastavení dovoluje změnu obrázku, názvu a také druhu rostliny z databáze. Mimo to je zde možné rostlinu „smazat“, čímž dojde k odpárování přiřazené rostliny a všech dalších vazeb. Přepnutím na přehled modulů v nejvyšší úrovni aplikace jsou do tabulky vykresleny jednotlivé moduly. Každý řádek obsahuje název modulu, jeho fotografii, čas od poslední synchronizace a případné upozornění na nízký stav vody v zásobníku. V detailu je potom navíc doplněno procentuální množství vody v zásobníku a vykresleny graficky čtyři „sloty“, odpovídající 4 zásuvkám a čerpadlům akčního modulu. Na každý slot může být přiřazen právě jeden senzor a tím dojde k jeho spárování. Párování s některým ze slotů lze provést při vytváření rostliny (bude popsáno níže), nebo v nastavení modulu, kde lze senzory ze slotů také odstraňovat. V nastavení je dále možná změna obrázku a vlastního názvu.

Přidání rostliny probíhá pomocí ikonky plus v pravém horním rohu aplikace a obsahuje následující kroky:



Obrázek 5.14: Detail senzoru, přidání (spárování) nové rostliny a detail druhu rostliny

1. Volitelný výběr vlastního obrázku a to jak z galerie, tak pomocí fotoaparátu. V případě kdy senzor nemá přiřazený vlastní obrázek, použije se obrázek zvoleného druhu rostliny.
2. Vlastní název rostliny. Pokud není vyplněn je použit výchozí název, složený vždy ze slova *Sensor* a číselného ID odpovídajícího přiřazené adrese v lokální síti.
3. Povinným parametrem je výběr druhu rostliny z databáze. Každá rostlina obsahuje specifické parametry pro svoje pěstování.
4. Dalším povinným polem je výběr fyzického senzoru, který má být s rostlinou spárován. Kliknutím uživatel dostane seznam všech dosud nespárovaných senzorů, které se v síti nacházejí.
5. Posledním volitelným krokem je spárování s akčním modulem. Zde se provádí výběr modulu a jednoho ze čtyř volných slotů na něm.

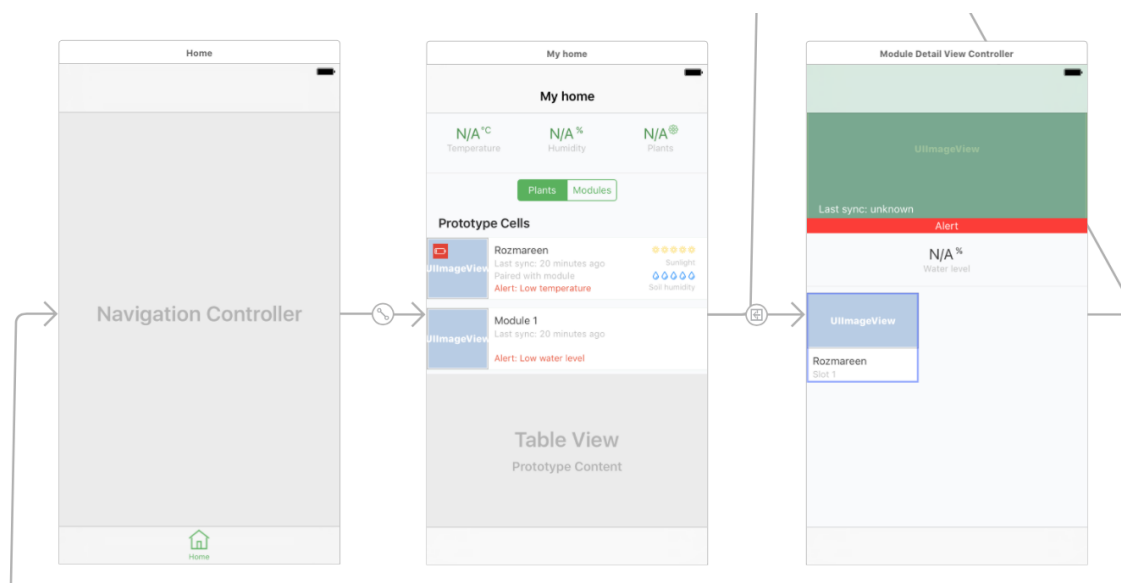
Vývoj uživatelského rozhraní probíhal v několika fázích. V první řadě byly vytvořeny náčrtky na papír pro prvotní přehled, jaké části bude aplikace obsahovat a jaké informace v nich budou obsaženy. Následně byl vytvořen prototyp aplikace, včetně prvního grafického návrhu. Při vývoji tak byl přeskočen krok tvorby wireframů, neboli česky „drátěných modelů“. Namísto toho byla větší pozornost soustředěna právě prototypu. Jednotlivé obrazovky byly nakresleny v grafickém nástroji Adobe Photoshop a následně „rozhýbány“ pomocí prototypovacího nástroje Marvel¹⁶. Ten umožňuje naskládat statické obrázky do

¹⁶Dostupné z: <https://marvelapp.com>

formy aplikace tak, jak budou ve výsledku vypadat. V obrázcích jsou definována místa, které mají vykonat určitou akci, nejčastěji v podobě přechodu na jinou obrazovku. Samotný styl přechodu lze také zvolit, aby odpovídal co nejvíce realitě. V prototypu se dá na obrazovce i pohybovat, stejně jako se dají vytvořit různé vlastní animace. Nástroj Marvel má svou mobilní aplikaci, díky čemuž lze prototyp testovat přímo na mobilním zařízení a dostat tak opravdu autentický zážitek. Takto vytvořený prototyp uživatelského rozhraní byl představen pár lidem, aby byla zjištěna zpětná vazba. Nutno podotknout, že bylo potřeba vybrané osoby nejprve seznámit s celým systémem a především jeho jednotlivými fyzickými částmi. V této fázi nebyly zjištěny žádné větší nedostatky a rozhraní tak mohlo být implementováno.

Realizace aplikace

Aplikace byla realizována v prostředí Xcode pod operačním systémem Mac OS, které je výchozím nástrojem pro tvorbu iOS aplikací. K samotnému vývoji je potřeba vývojářský účet, jenž je placený, ale VUT jej má k dispozici pro studenty zdarma. Jediným rozdílem oproti placené verzi je nemožnost výslednou aplikaci publikovat do obchodu *AppStore*. Jako programovací jazyk byl zvolen Swift v jeho poslední verzi 3.1. Aplikace využívá několika knihoven třetích stran. V první řadě jsou to knihovny pro práci se službou *Firestore*, dále knihovna pro kreslení grafů *Charts*¹⁷, knihovna *AlamofireImage*¹⁸ pro stahování obrázků a jejich ukládání do dočasné paměti (cache) a další menší knihovny. Pro jejich správu bylo využito nástroje *CocoaPods*¹⁹, který obsahuje více než 31 mil. knihoven.



Obrázek 5.15: Ukázka uživatelského rozhraní v nástroji Storyboard

Struktura aplikace je rozdělena přesně podle architektury MVC (*Model-view-controller*). Část modelu je tvořena jednotlivými třídami reflektujícími datový model systému. Každá třída obsahuje svůj „service“ (např. třída *Sensor* má k sobě *SensorService*). Servisy tvoří

¹⁷Dostupné z: <https://github.com/danielgindi/Charts>

¹⁸Dostupné z: <https://github.com/Alamofire/AlamofireImage>

¹⁹Více informací: <https://cocoapods.org>

spojovací vrstvu mezi částí modelu a rozhraní. Provádí se zde načítání dat z realtime databáze služby Firebase a změny jednotlivých atributů. Firebase rozlišuje načítání dat dvojnásobným způsobem. V prvním případě metodou `observeSingleEvent()` jsou staženy data pouze jedenkrát a z funkce vráceny pomocí „completion“ bloku. To je speciální konstrukce, kdy v místě volání funkce se definuje blok kódu, který se má vykonat v případě kdy je completion blok zavolán. Jelikož načítání dat z cloudové databáze je časově náročná operace, provede se zavolání funkce představující požadavek na stažení dat a jakmile jsou data dostupná, zavolá se completion blok se staženými daty a v místě volání se provede jejich vypsání. Naopak metodou `observe()` je stažení dat z realtime databáze provedeno vždy při změně některé hodnoty. Completion block je tak volán opakovaně, čehož je využito na většině míst v aplikaci a tím jsou zobrazená data vždy aktuální. Completion blok provádí nejčastěji překreslení tabulky či seznamu s daty.

Uživatelské rozhraní (*View*) je vytvořeno pomocí nástroje *Storyboard*. Ten umožňuje grafickou tvorbu rozhraní ze standardních iOS komponent a definování vazeb mezi jednotlivými obrazovkami. Ukázkou části takto vytvořeného uživatelského rozhraní lze vidět na obrázku 5.15. Jednotlivé obrazovky mají definované velikosti a vazby mezi komponentami pomocí nástroje *AutoLayout*, díky čemuž je rozhraní univerzální pro libovolnou velikost displeje. Část rozhraní, konkrétně zobrazení optimálních parametrů pro pěstování jednotlivých druhů rostlin v databázi, bylo realizováno ručně v kódu, namísto grafického editoru. Každá obrazovka má přiřazený svůj *Controller*, tedy třídu, která definuje její chování. V těchto třídách se nejčastěji provádí požadavek na načtení dat, které mají být vykresleny, úprava grafického vzhledu, či definování akcí na základě vzniklých událostí (např. stisknutí tlačítka).

Kapitola 6

Testování a finanční zhodnocení

V této kapitole jsou shrnuty způsoby testování bezdrátové komunikace, jednotlivých modulů i celého systému jako celku. Dále je zde shrnuta finanční stránka projektu, jelikož jedním z nepsaných cílů bylo vytvořit systém levně.

6.1 Test bezdrátové komunikace

Nejprve bylo nutné otestovat vybranou bezdrátovou technologii a změřit její dosah, zda odpovídá požadavkům práce. K tomu byly vytvořeny jednoduché moduly obsahující řídicí desku Arduino Nano, komunikační modul NRF24L01 a teplotní čidlo Dallas DS18B20. U modulů NRF24L01 lze zvolit jeden ze 4 vysílacích výkonů, jak již bylo v práci zmiňováno. Testování probíhalo v bytě o velikosti 110m² a komunikace byla úspěšná ve všech jeho částech, při nastavení nejvyššího vysílacího výkonu. V případě druhého nejnižšího výkonu, který byl zvolen pro bateriově napájený modul, byla komunikace úspěšná na vzdálenost 10m skrze tři nosné zdi. Při testování byly použity moduly s integrovanou anténou a pro centrální uzel modul s anténou externí. Důležitým faktorem je také výběr konkrétní rádiové frekvence. V první fázi byla její hodnota ponechána na 2,497GHz, což je výchozí hodnota zvolená v knihovně RF24Mesh.

Dosah bezdrátové komunikace byl posléze ještě ověřen v rámci závěrečného testování. Na půdních senzorech jsou totiž použity komunikační moduly v SMD verzi, které mají údajně horší anténu a menší dosah. Při testování se ukázal tento fakt jako pravdivý. Klasické moduly s věstavenou anténou dosahovali zmíněných 10m, avšak jejich SMD verze pouze zhruba 7m skrze jednu zeď. V reálné síti to ale nepředstavuje velký problém, jelikož se předpokládá umístění půdních senzorů v blízkosti akčních modulů, které v rámci mesh topologie mohou data přeposílat do centrálního uzlu. Akční moduly jsou neustále napájeny, jejich komunikační modul obsahuje lepší anténu a také je zde zvolen nejvyšší vysílací výkon. Komunikace byla primárně testována mezi vysílacími uzly a centrálním modulem, ke kterému se připojují. Zjištěné dosahy v domácích podmínkách jsou uvedeny v tabulce 6.1. Výsledky jsou pouze orientační a skutečná hodnota závisí vždy na okolních podmínkách. Pokud by bylo měření prováděno například v centru města, kde jsou používaná bezdrátová pásma velmi hustě obsazena, výsledky mohou být odlišné. Do tabulky byly zaznamenány ty hodnoty, při kterých byla komunikace ve většině případů stabilní. Maximální dosah odpovídá zhruba dosahu lokální Wi-Fi sítě v testovaném bytě.

V této fázi testování byly také zkoumány různé vysílací frekvence. Ačkoliv se předpokládalo, že na vyšších frekvencích bude docházet k menšímu rušení, opak byl pravdou. Nej-

Uzel	Typ antény	Podmínky	Dosah
Akční modul	Vestavěná	Několik nosných zdí	15m
Půdní senzor - prototyp	Vestavěná	Tři nosné zdi	10m
Půdní senzor	Vestavěná SMD	Jedna nosná zeď	7m

Tabulka 6.1: Zjištěný maximální dosah bezdrátové komunikace v domácích podmínkách

vhodnější volbou se tak staly frekvence kolem výchozí hodnoty 2,497GHz. V praxi se občas také využívá manuálního přeskakování frekvencí, jako u metody FHSS (*Frequency Hopping Spread Spectrum*). Manuální proto, že moduly komunikují primárně na jedné frekvenci, ale uživatel ji může sám v průběhu měnit. Uzly se snaží navázat komunikaci na výchozí frekvenci a pokud selže, zkusí to na frekvencích jiných. Ty ale musí být dohodnuty předem. Příjemce průběžně monitoruje všechny dohodnuté frekvence a snaží se data přimout. Při testování se ale neukázaly větší problémy se selháním komunikace a tak byla zvolena pouze jedna frekvence a to 2,498GHz. Závěrečné testování probíhalo kromě výše zmíněného bytu také v rodinném domě o dvou patrech a rozloze asi 250m². Zde již byly zjištěny drobné výpadky spojení i při nejvyšším vysílacím výkonu na vzdálenost 15m skrze jedno patro a několik zdí. Nutno podotknout, že pro finální testování byly použity již hotové moduly, kdy anténa je uzavřena uvnitř plastových krabiček, které dosah také lehce snižují. Celkově je bezdrátová technologie pomocí modulů NRF24L01 spolehlivá pro použití ve většině domácích podmínek.

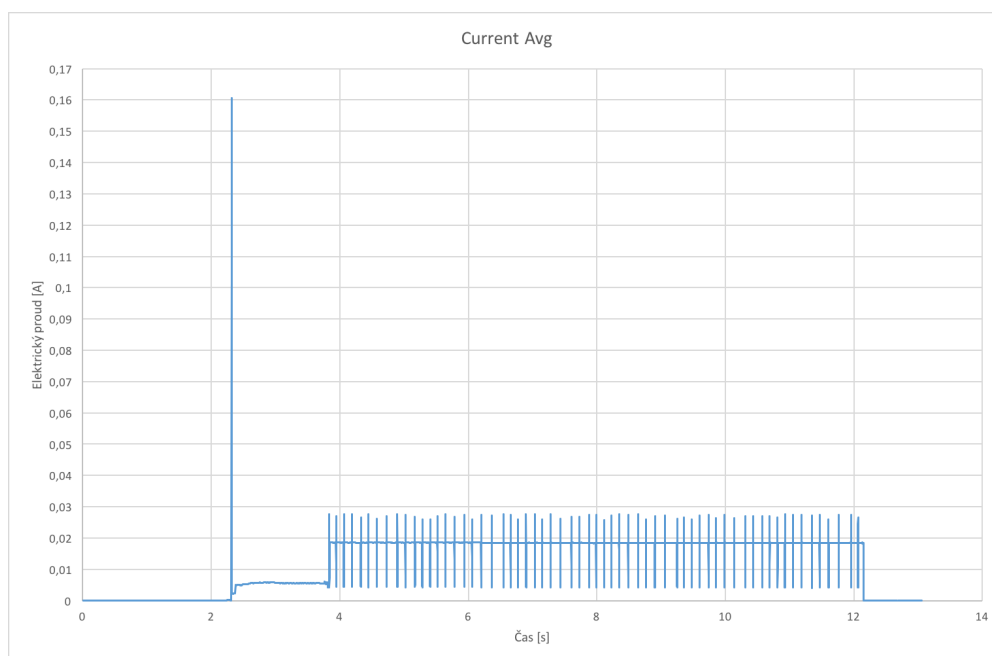
6.2 Testování modulů a celého systému

Jednotlivé moduly byly testovány vždy hned po jejich sestavení. Testovala se především bezchybnost dlouhodobého běhu, schopnost bezdrátové komunikace, měření dat ze senzorů a vykonávání událostí na akčních prvcích. Při závěrečném ladění programů, podle zjištěných nedostatků, se ukázalo jako velmi nepraktické ruční zadávání MAC adresy každého uzlu na začátek programu. Byla proto zvolena metoda jiná, kdy je nejprve na mikroprocesor uložena přiřazená adresa do nevolatilní paměti EEPROM na prvních 20 bajtů a následně až nahrán samotný řídicí kód. Adresa tak zůstává v paměti stále uložena, nemění se, řídicí program ji pouze čte a sám pracuje s EEPROM pamětí až od adresy 20. Pro lepší přehlednost byl každý uzel obstarán štítkem se svojí MAC adresou.

Gateway modul, sestavený v počtu jeden kus, byl úspěšně testován po dobu 14 dní, kdy běžel nepřetržitě bez jediných anomálií. Čas od času vlivem hardwarového watchdog časovače dojde k resetování řídicího modulu NodeMCU, ale po resetu je modul opět uveden do plně funkčního stavu. Drobným problémem se zde ukázalo měření teploty pomocí senzoru HTU21D. Ten se nachází uvnitř krabičky spolu s ostatními komponentami, mezi kterými je i lineární regulátor napětí z 9V na 5V. Regulátor produkuje mírné teplo dostatečné pro vytopení krabičky, čímž je způsobena nepřesnost oproti skutečné teplotě. Měřením se ukázala odchylka asi 2-3°C. Možným řešením by mohlo být umístění teplotního senzoru mimo krabičku, použití přímo 5V napájecího napětí bez nutnosti regulátoru, nebo programová kompenzace chyby.

Jako další komponenta byl testován **půdní senzor**, který je vyroben v jednom kusu jako prototyp a celkem 5 kusech na vlastní desce plošných spojů. Nejprve byly vyhodnocovány naměřené hodnoty ze senzorů na prototypu a z nich určeny například krajní hodnoty pro vlhkou půdu, suchou půdu, apod. Větší pozornost byla věnována testování probouzecího

mechanizmu výsledných desek pomocí Nano timerů TPL5110. Doba probouzení je určena zvoleným odporem, kdy 2 senzory jsou nastaveny zhruba na 1 hodinu, 2 na 30 minut a jeden na 15 minut. Naměřené data jsou zapisována do cloudové databáze, kde lze zjistit, zda opravdu došlo k odeslání v nastavených intervalech. V této fázi byly objeveny problémy s doručováním dat, jak bylo popsáno u realizační části. Po rozdělení zasílaných zpráv na dvě, namísto jedné větší, byly výpadky velmi řídké. Sensor s nastaveným intervalem probouzení na 15 minut se probouzí opravdu vždy po 15 minutách $\pm 1s$. U delších intervalů již vznikají větší nepřesnosti až v rámci minut. Průměrná doba běhu po probuzení senzoru byla zjištěna měřením asi 5s, maximální doba potom 8s v případě, kdy se sensor nepodaří připojit do sítě a je uspán časovačem. Známým problémem při měření půdní vlhkosti je ztráta kovu na elektrodách senzoru způsobená elektrolýzou. Tento problém se dá řešit napájením pomocí střídavého proudu, což je v případě již hotového senzoru, jako v této práci, problematické, nebo spínáním napětí po velmi krátkou dobu. I když je doba běhu maximálně 8s a tedy sensor je napájen maximálně po tuto dobu, byly po týdenním testování pozorovány velmi mírné ztráty kovu na elektrodách. Do budoucna by tak stálo za zvážení zapojení tranzistoru jako spínače pro sensor půdní vlhkosti, nebo jej přímo integrovat do desky plošných spojů.



Obrázek 6.1: Průměrný odběr elektrického proudu půdního senzoru

U půdního senzoru bylo také provedeno **měření jeho proudového odběru**, aby bylo možné orientačně odhadnout životnost na knoflíkovou baterii CR2032. Měření probíhalo pomocí přístroje DC Power Analyzer od společnosti Keysight. Výsledky byly zaznamenávány do souboru a následně vyneseny do grafu 6.1. V čase 2,2s bylo manuálně sepnuto napájení obvodu pomocí START tlačítka. V grafu lze pozorovat krátkodobou špičku, kdy došlo pravděpodobně k nabytí kondenzátorů a proudový odběr tak činil 0,16A. Poté proběhla inicializace mikroprocesoru po dobu 1s, během které má obvod odběr zhruba 5mA. Následuje 8s interval navazování bezdrátové komunikace a poté uspání obvodu pomocí Nano timeru. Průměrný odběr při bezdrátové komunikaci je tak zhruba 17-18mA. V režimu spánku byl změřen proud menší než $1\mu A$, což je nejmenší rozlišovací schopnost zmíněného přístroje. Nano Timer TPL5110 má udávanou spotřebu v klidovém stavu 30nA. K té je ale

potřeba ještě připočítat malý proud, který protéká vypnutým tranzistorem. Konkrétně pro P-MOSFET tranzistor použitý na deskách je to dle technické dokumentace $\pm 10\mu\text{A}$. Z naměřených údajů a kapacity baterie CR2032 220mAh bylo vypočítáno, že v případě měření každých 30 minut je životnost asi 180 dní, tedy půl roku. Pokud by docházelo k probouzení jednou za hodinu, životnost by se tak zvýšila až na jeden rok. Pro výpočty byl uvažován téměř maximální odběr po průměrnou dobu běhu 5s. Vlivem výpadků v komunikaci, kdy je senzor napájen až 8s, by se životnost baterie mohla lehce snížit.



(a) Půdní senzor a zalévání

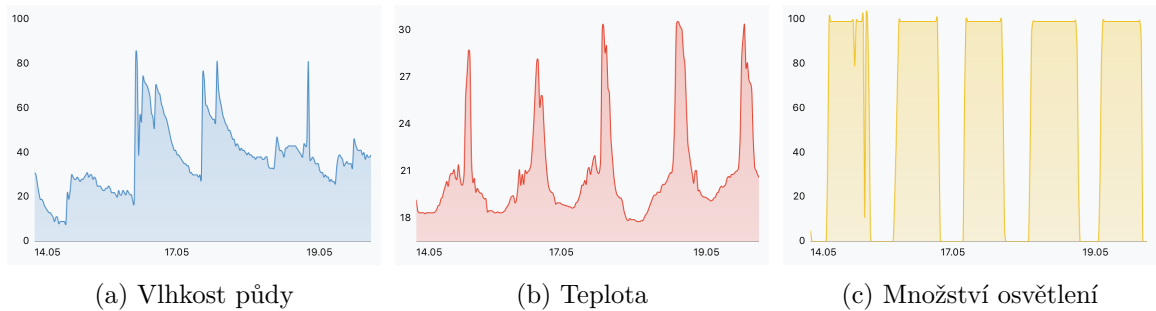


(b) Umělé osvětlení

Obrázek 6.2: Testování automatického systému

Akční modul byl testován především v závěrečné fázi, kdy tak byla kontrolována logika celého systému. Tedy kromě samotné funkčnosti spínání/vypínání relé a spínání čerpadel po definovaný čas, bylo testováno, zda opravdu dochází k automatickému řízení na základě dat vyhodnocených ze senzorů. Pro tyto účely byl k modulu spárován půdní senzor a vše nainstalováno do květináče, kde systém běžel po několik dní. Ukázky z testování lze vidět na obrázku 6.2. Při této fázi se projevil drobné nedostatky, na které bylo již při návrhu myšleno, že by mohli nastat. Prvním problémem jsou použité ponorné čerpadla, které nemají dostatečný výkon, aby byla schopná vyčerpat vodu do výšky více než 1m. Podle jejich parametrů mají maximální zdvih vody 40-110cm a ačkoliv při prvotním testování byly dosahovány výsledky kolem 1m, ve výsledné realizaci, kdy je čerpaná voda vedena několika hadicemi s různými průměry skrze redukce, byl zdvih okolo 75cm. Druhý problém způsobuje zemská gravitace. V případě, že je zalévací box umístěn výše než samotné květináče, dochází vlivem gravitace ke snaze o vyrovnání hladin a tím pádem k samovolnému zalévání, které je nežádoucí. Na tento problém bylo v průběhu vývoje myšleno a tak byly zakoupeny elektricky ovladatelné vodní ventily, ovšem k jejich nasazení nedošlo. Samy ventily jsou relativně drahou položkou oproti čerpadlům, čímž by se značně zvýšila cena celého modulu. Do budoucna by bylo vhodnější použití jednoho výkonnějšího čerpadla a několika ventilů i za vyšší cenu. Pro správné fungování je tak potřeba umístit zásobník nejlépe na stejnou úroveň jako květináče, případně níže. Na grafech 6.3, získaných z mobilní aplikace, lze vidět naměřené výsledky ze senzorů po dobu 5 dní. Na prvním (6.3a) je zobrazena vlhkost půdy a lze zde pozorovat místa, kdy došlo k zalévání v podobě skokového zvýšení hodnoty, následované postupným poklesem. Prostřední graf (6.3b) vyjadřuje vývoj teploty a na posledním grafu 6.3c je množství dopadajícího světla. V jarním období bylo denní světlo detekováno většinou svojí plnou hodnotou, a proto lze v grafu vidět obrovské změny z minima do ma-

xima a naopak, jak slunce vycházelo a zase zapadalo. Vlhkost půdy a množství světla je v grafu vyneseno jako procentuální hodnota od 0-100%, teplota potom ve stupních Celsia.



Obrázek 6.3: Grafy naměřených hodnot z půdního senzoru

6.3 Finanční zhodnocení

Jedním z osobních cílů projektu byla celková cena. Podobné systémy se dají na trhu v různých formách zakoupit, ovšem téměř vždy vynikají velmi vysokou cenou. Následuje proto orientační cenové zhodnocení vyrobených prototypů. Je zde počítána pouze cena materiálu, bez vlastní práce, na jeden kus. Většina řídicích modulů, mikroprocesory, senzory apod. byla zakoupena v Číně, krabičky, drobné elektronické součástky, konektory atd. potom v tuzemských obchodech a některé komponenty byly zakoupeny také z Anglie a Francie.

- **Gateway modul:**

Položka	Cena [Kč]
Řídicí modul NodeMCU	80,-
Komunikační modul NRF24L01 ve verzi s externí anténou	65,-
Senzor teploty a vlhkosti HTU21D	57,-
Drobná elektronika, konektory apod.	cca. 30,-
Plastová krabička	152,-
Celkem	cca. 384,-

- **Půdní senzor:**

Položka	Cena [Kč]
Výroba desky plošných spojů	33,-
Mikroprocesor ATmega328p-AU	31,-
Komunikační modul NRF24L01 SMD verze	41,-
Teplotní čidlo Dallas DS18B20	16,-
Fotorezistor	5,-
Senzor půdní vlhkosti	12,-
Nano timer TPL5110	22,-
Drobná elektronika, konektory apod.	cca. 30,-
3D tisk plastové krabičky	600,-
Celkem bez 3D tisku	cca. 190,-
Celkem	cca. 790,-

- **Zásobník vody:**

Položka	Cena [Kč]
Ponorné čerpadla 4ks	145,-
Ultrazvukový senzor HC-SR04	48,-
H-můstek 2ks	75,-
Drobná elektronika, konektory apod.	cca. 30,-
Malá plastová průmyslová krabička	73,-
Odpadkový koš jako zásobník	99,-
Celkem	cca. 475,-

- **Řídící box:**

Položka	Cena [Kč]
Arduino Nano	85,-
Relé deska (4 relé)	53,-
5V regulátor	30,-
Drobná elektronika, konektory apod.	cca. 70,-
Plastová průmyslová krabička	169,-
Zásuvky 4ks	300,-
Celkem	cca. 707,-

Z výše uvedených tabulek je vidět, že ani výroba vlastního systému nebyla úplně nejlevnější. Půdní senzor byl navíc vyroben celkem v 5 funkčních kusech, ale pouze s jednou krabičkou, jelikož její 3D tisk byl příliš nákladný. Celková cena systému se tak vyšplhala přes 3000Kč + asi 1000Kč na ostatní komponenty, které nebyly použity pro výsledný návrh. Jedná se ovšem o prototypovou výrobu, případná sériová výroba by byla podstatně levnější. Nejdražší položky jsou tvořeny plastovými krabičkami a dalšími fyzickými komponentami. Paradoxně je tak elektronika v nich obsažená mnohonásobně levnější.

Kapitola 7

Závěr

Cílem této práce bylo navrhnout a realizovat systém pro automatizované pěstování domácích rostlin připojený do Cloudu se vzdáleným přístupem pomocí mobilní aplikace.

K dosažení bylo potřeba nastudovat problematiku Internetu věcí a zpracování dat v Cloudu. Více se tato práce zabývá konkrétním využitím v domácí automatizaci pro pěstování rostlin pomocí moderních technologií. Jsou zde představeny vybrané hydroponické a závlahové systémy, které se v blízké době vyskytnou, nebo se již vyskytují na trhu. K realizaci konkrétního chytrého hardwaru a jeho připojení do internetu věcí byly představeny různé technologie bezdrátové komunikace, vybrané komunikační protokoly a představena řada hardwarových komponent pro stavbu chytrých zařízení. Z nich byly navrženy konkrétní moduly, které byly realizovány jak po hardwarové stránce, tak i softwarové. Konkrétně se jedná o gateway modul, zajišťující správu lokální bezdrátové sítě a komunikaci s cloudovými službami, půdní senzory pro měření vlhkosti půdy, teploty a množství osvětlení v květináčích a akční modul pro automatizované zalévání a umělé osvětlení rostlin. K půdnímu senzoru byl vytvořen navíc jeho kompletní 3D model. Značná část práce je věnována hardwarovému návrhu elektronických obvodů, řídicích jednotek a fyzickému umístění těchto jednotek do krabiček a boxů. V rámci práce vznikla také knihovna pro párování uzlů v lokální bezdrátové síti s mesh topologií vytvořené pomocí modulů NRF24L01, která je obecně použitelná a rozšiřitelná i pro jiné oblasti chytrých domácností. Celý systém byl skrze Wi-Fi modul připojen k internetu a cloudové službě Google Firebase, kde byla realizována celá řídicí logika. Tato část by mohla být do budoucna více rozšířena za využití ostatních cloudových služeb, které společnost Google nabízí a nad naměřenými daty tak provádět detailnější analýzy. Jako poslední část systému vznikla nativní mobilní aplikace pro platformu iOS napsaná v jazyce Swift 3.1.

Celý systém se podařilo úspěšně realizovat ve formě funkčního a plně použitelného prototypu, čímž byly všechny body zadání splněny. V realizační části práce jsou popsány možná vylepšení a změny, které by mohly být v budoucnu provedeny. Největším nedostatkem je pravděpodobně realizace akčního modulu, který z důvodu bezpečnosti musel být rozdělen na dvě části – zásobník s vodou a akční box. Tím se kazí vizuální dojem z celého systému. Gateway modul i půdní senzory jsou umístěny v hezkých krabičkách, zatímco akční modul v krabičkách průmyslových. Pokud by měl být systém v nějaké formě dále rozvíjen, muselo by se více investovat do vývoje speciálního boxu spojujícího oddělený zásobník vody a silnoproudou část. Takto vytvořený box by navíc musel projít testy na voděodolnost a celkovou bezpečnost. Také navržené a realizované desky plošných spojů pro půdní senzor by mohli být ještě vylepšeny o integrovaný senzor půdní vlhkosti. V průběhu vývoje byl průběžně zjišťován zájem o systém a zda by jej lidé doma využili. Zajímavým zjištěním bylo,

že více než polovina dotázaných by si systém raději půjčila například po dobu dovolené, než zakoupila a aktivně používala po celý rok.

Vytvořená práce má několik přínosů. V první řadě je to celkové seznámení s novou problematikou internetu věcí a chytrých domácností, nastudování hardwarového návrhu zařízení spadajících do dané oblasti a také manuální zručnost v podobě pájení a osazování vlastních desek plošných spojů. Vytvořená párovací a komunikační knihovna je navíc lehce rozšiřitelná pro libovolné použití. Není tak vyloučeno, že bezdrátová síť nebude v budoucnu rozšířena například o uzly zajišťující termoregulaci v bytě, automatické osvětlení či monitorující bezpečnost. Zda bude systém dále vyvíjen v aktuální podobě zatím není zcela jasné. Minimálně vytvořený a funkční prototyp bude používán, testován a nadále vylepšován pro osobní potřeby.

Literatura

- [1] *Bluetooth*. [Online; navštíveno 28.12.2016].
URL <https://www.bluetooth.com>
- [2] *Wi-Fi Alliance – Discover Wi-Fi*. [Online; navštíveno 29.12.2016].
URL <http://www.wi-fi.org/discover-wi-fi>
- [3] *Wi-Fi Alliance – Wi-Fi Direct*. [Online; navštíveno 29.12.2016].
URL <http://www.wi-fi.org/discover-wi-fi/wi-fi-direct>
- [4] *ZigBee Alliance – Global leaders creating innovative products and services for the Internet of Things*. [Online; navštíveno 29.12.2016].
URL <http://www.zigbee.org/zigbeealliance/our-members/>
- [5] Apple Inc.: *The Swift Programming Language*. online, 2014.
URL <https://itunes.apple.com/cz/book/swift-programming-language/id881256329?mt=11>
- [6] Banks, A.; Gupta, R.: *MQTT Version 3.1.1 Plus Errata 01*. Technická zpráva, December 2015, [Online; navštíveno 27.12.2016].
URL <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html>
- [7] Belshe, M.; Peon, R.; Thomson, M.: *Hypertext Transfer Protocol Version 2 (HTTP/2)*. RFC 7540, May 2015, [Online; navštíveno 25.12.2016].
URL <http://www.rfc-editor.org/rfc/rfc7540.txt>
- [8] Faludi, R.: *Building Wireless Sensor Networks*. Sebastopol, CA: O'Reilly Media, Inc, December 2010, ISBN 978-0-596-80773-3.
- [9] Fielding, R. T.; Gettys, J.; Mogul, J. C.; aj.: *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616, June 1999, [Online; navštíveno 25.12.2016].
URL <http://www.rfc-editor.org/rfc/rfc2616.txt>
- [10] Friess, P.: *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*. River Publishers, 2013, ISBN 978-8792982735.
- [11] Griffith, E.: *What Is Cloud Computing?* Květen 2016, [Online; navštíveno 23.12.2016].
URL <http://www.pcmag.com/article2/0,2817,2372163,00.asp>
- [12] IoT Portál brána do světa internetu věcí: Využití. [Online; navštíveno 22.12.2016].
URL <http://www.iot-portal.cz/vyuziti/>

- [13] McEwen, A.; Cassimally, H.: *Designing the internet of things*. Chichester, United Kingdom: Wiley, 2014, ISBN 978-1-118-43062-0.
- [14] Minerva, R.; Biru, A.; Rotondi, D.: *Towards a definition of the Internet of Things (IoT)*. Telecom Italia S.p.A., Květen 2015, [Online; navštíveno 22.12.2016].
URL http://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf
- [15] Palivec, P.: *Bluetooth Low Energy*. March 2012, [Online; navštíveno 28.12.2016].
URL <http://www.dps-az.cz/soucasky/id:9912/bluetooth-low-energy>
- [16] Pfister, C.: *Getting Started with the Internet of Things*. Sebastopol, CA: O'Reilly Media, Inc, May 2011, ISBN 978-1-4493-9357-1.
- [17] Redakce HW serveru: *ZigBee – novinka na poli bezdrátové komunikace*. July 2005, [Online; navštíveno 29.12.2016].
URL <http://vyvoj.hw.cz/navrh-obvodu/rozhrani/zigbee-novinka-na-poli-bezdratove-komunikace.html>
- [18] Roberto, K. F.: *How-To Hydroponics*. Farmingdale, New York: Future Garden Inc, 2000, ISBN 0-9672026-0-4.
- [19] Shelby, Z.; Hartke, K.; Bormann, C.: *The Constrained Application Protocol (CoAP)*. RFC 7252, June 2014, [Online; navštíveno 26.12.2016].
URL <http://www.rfc-editor.org/rfc/rfc7252.txt>
- [20] Stanford-Clark, A.; Truong, H. L.: *MQTT For Sensor Networks (MQTT-SN)*. November 2013, [Online; navštíveno 27.12.2016].
URL http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf
- [21] U.S. Department of Agriculture: *Urban Agriculture Toolkit*. [Online; navštíveno 24.12.2016].
URL <http://www.usda.gov/documents/urban-agriculture-toolkit.pdf>
- [22] Wikipedia: *Internet of things*. [Online; navštíveno 22.12.2016].
URL https://en.wikipedia.org/wiki/Internet_of_things

Přílohy

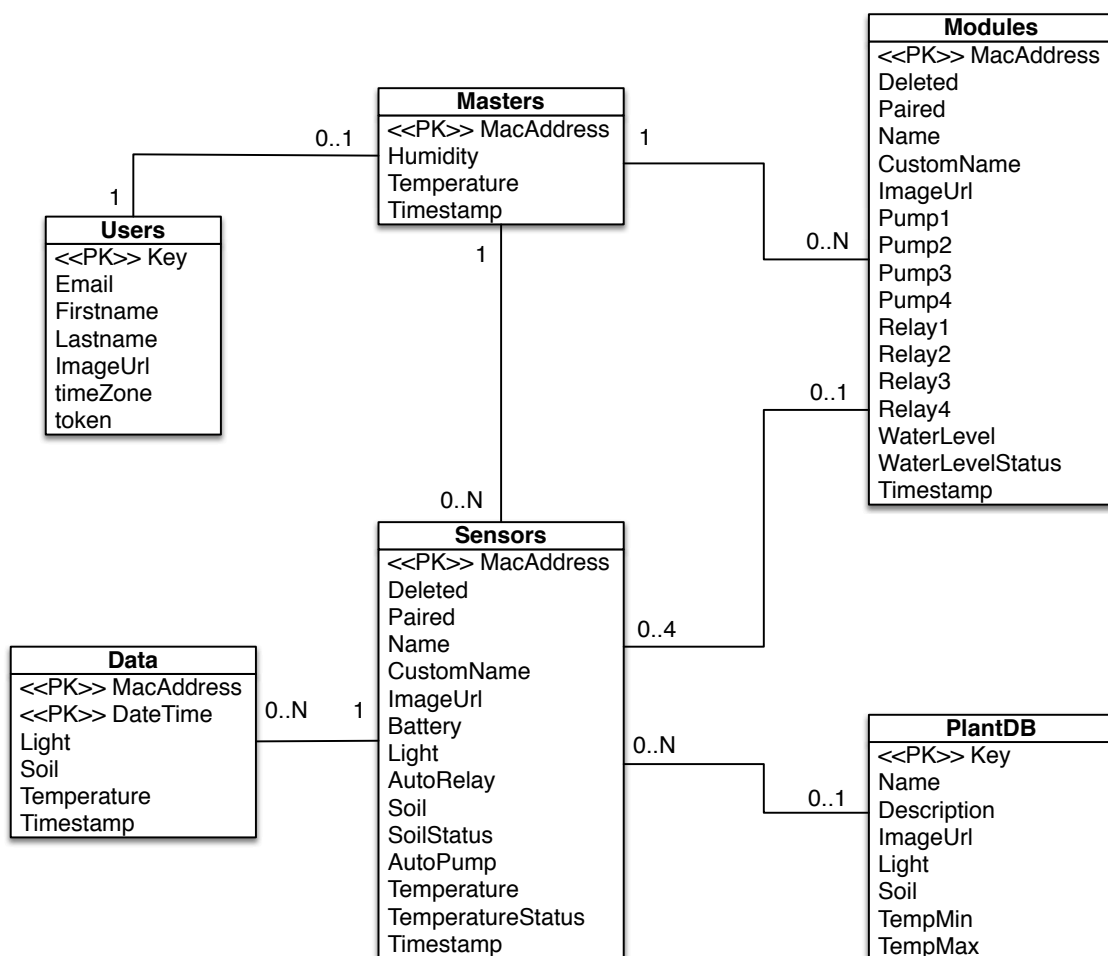
Příloha A

Obsah přiloženého paměťového média

- **/README.txt** – soubor s informacemi o obsahu paměťového média
- **/App** – mobilní aplikace, projekt programu Xcode
- **/Arduino** – řídicí programy v jazyce Arduino pro jednotlivé moduly
- **/Arduino/Libraries** – vlastní Arduino knihovny; modifikovaná knihovna *RF24Mesh*
- **/Cloud/functions** – logika systému pro Firebase Functions
- **/Cloud/initPlantDB.sh** – inicializační skript pro naplnění databáze druhů rostlin
- **/Měření** – výsledky měření proudového odběru půdního senzoru
- **/Model** – 3D model půdního senzoru
- **/Obrázky** – obrázky vyhotovených modulů, obrazovek aplikace a testování
- **/Schema** – elektrická schémata v nástroji Eagle
- **/TZ** – technická zpráva ve formátu PDF
- **/TZ/src** – zdrojové soubory technické zprávy v \LaTeX u

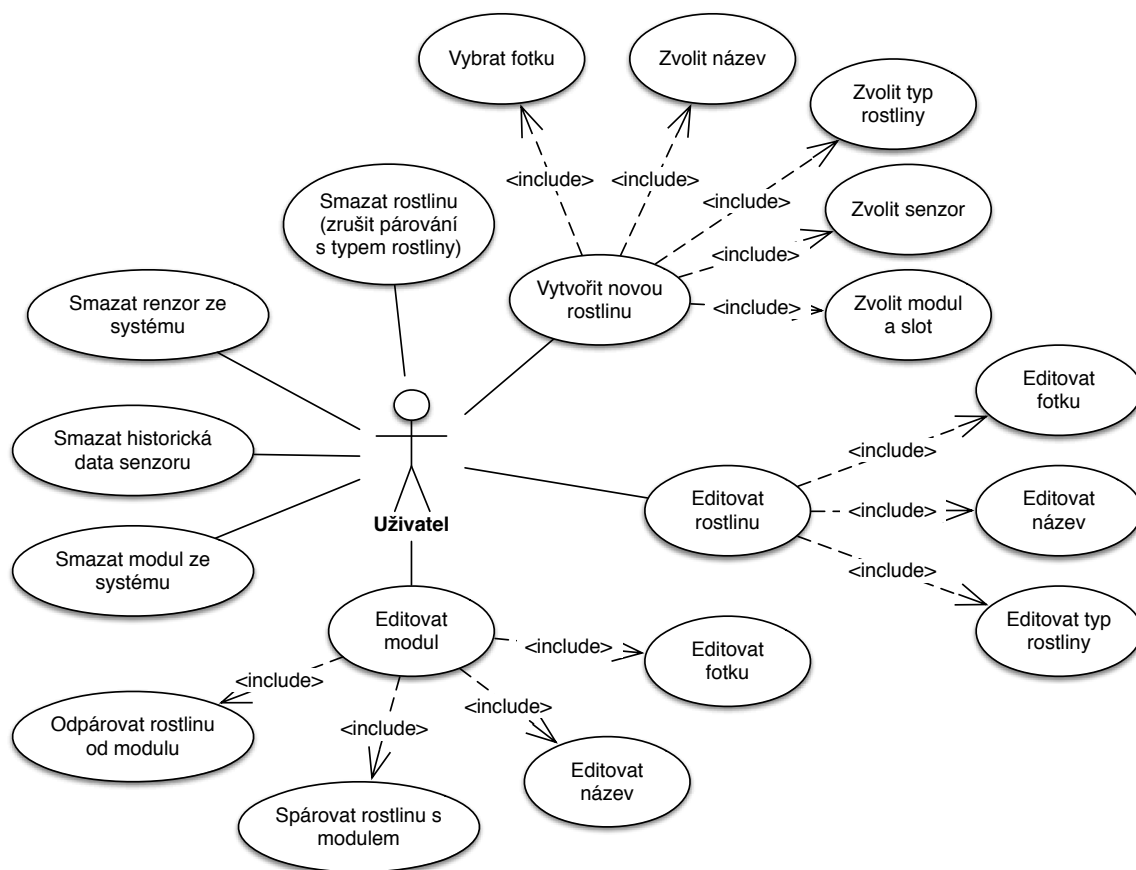
Příloha B

ER diagram datového modelu



Příloha C

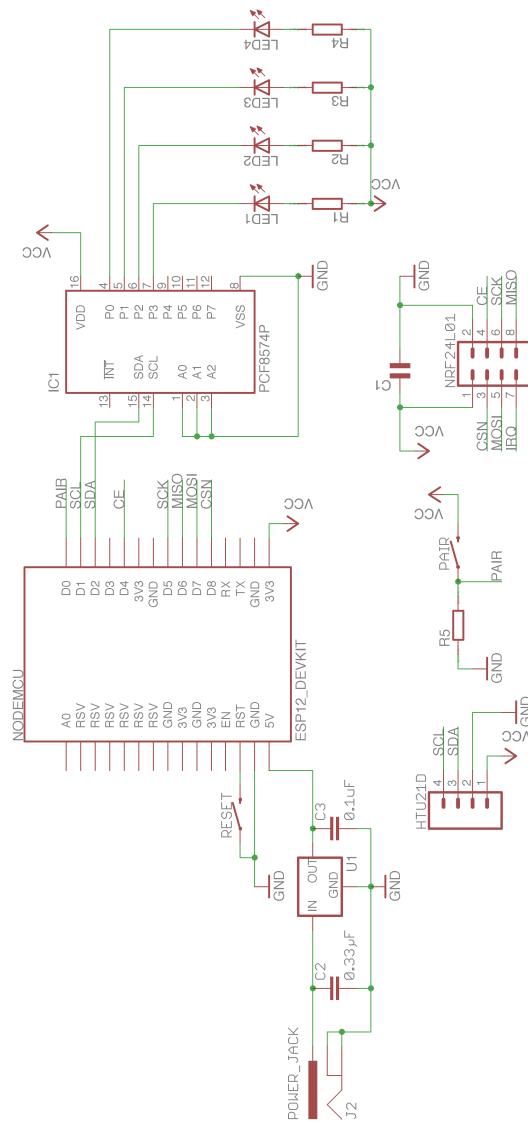
Diagram případů užití mobilní aplikace



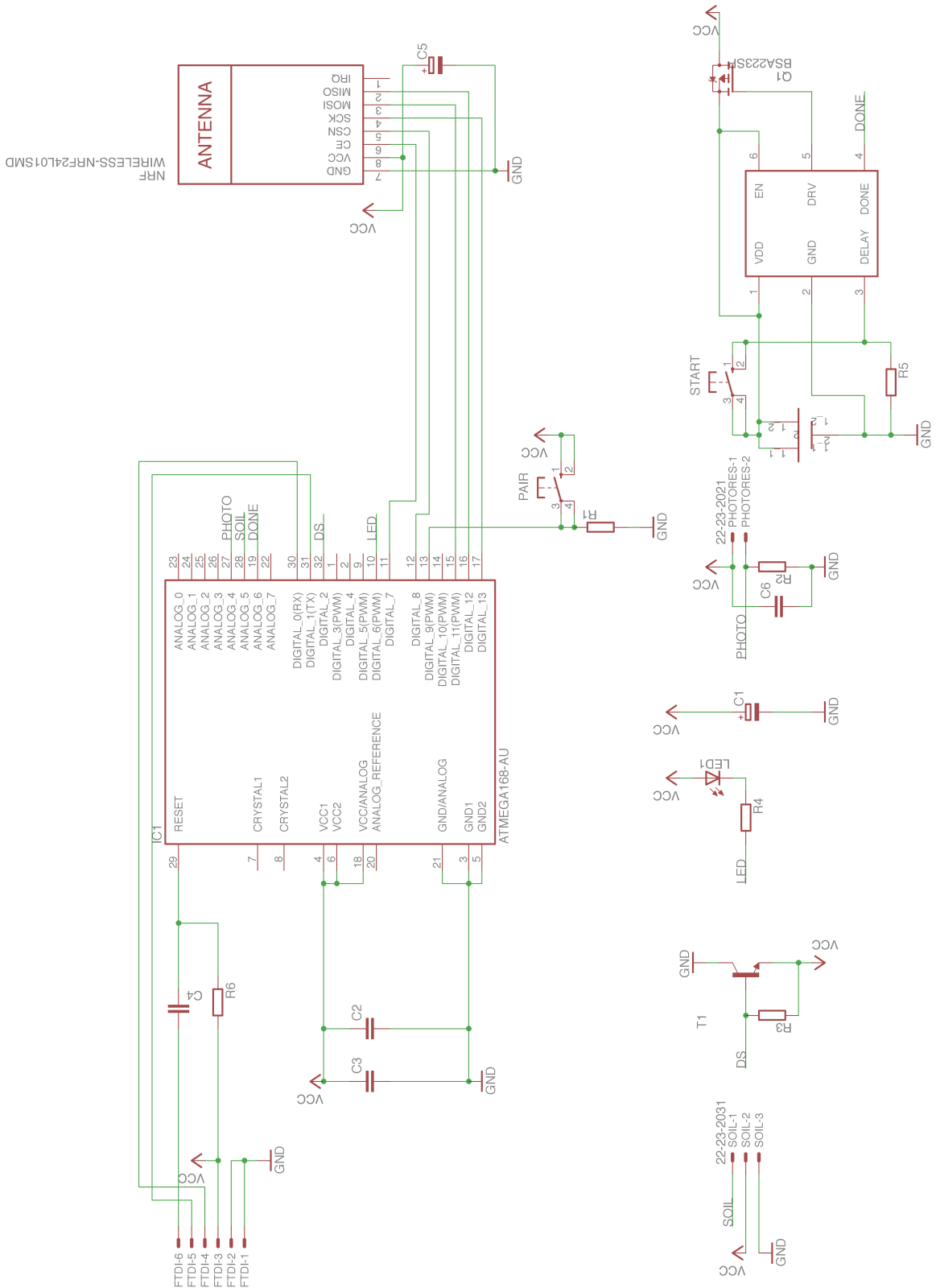
Příloha D

Elektrická schémata

D.1 Gateway modul



D.2 Půdní senzor



Příloha E

Podklady pro výrobu desky plošných spojů

