



## Zadání bakalářské práce

<b>Název:</b>	Chytrý květináč
<b>Student:</b>	Mikuláš Veselý
<b>Vedoucí:</b>	Ing. Martin Daňhel, Ph.D.
<b>Studijní program:</b>	Informatika
<b>Obor / specializace:</b>	Počítačové inženýrství
<b>Katedra:</b>	Katedra číslicového návrhu
<b>Platnost zadání:</b>	do konce letního semestru 2021/2022

### Pokyny pro vypracování

Analyzujte současné trendy v oblasti chytrých květináčů a podporovaných bylin.

Dle analýzy vyberte vhodný chytrý květináč a konkrétní rostlinu tak, aby bylo možné provést testování za účelem pozdějšího návrhu vlastního květináče na základě reverzního inženýrství (vývoj vlastního květináče není součástí této práce). Navrhněte a vytvořte testovací zařízení (platformu zvolte dle vlastního uvážení), pro které vyberte vhodný HW (řídící elektroniku, senzory teploty, vlhkosti, světla apod.).

Pro ukládání a vizualizaci naměřených provozních dat navrhněte a implementujte vhodnou webovou aplikaci, která bude umožňovat zobrazení vybraných statistik (tabulky, grafy apod.) ohledně teploty, vlhkosti půdy apod.



Bakalářská práce

# CHYTRÝ KVĚTINÁČ

Mikuláš Veselý

Fakulta informačních technologií ČVUT v Praze  
Katedra číslicového návrhu  
Vedoucí: Ing. Martin Daňhel, Ph.D.  
27. června 2021

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2021 Mikuláš Veselý. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bez uplatněných zákonných licencí nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.*

Odkaz na tuto práci: Mikuláš Veselý. *Chytrý květináč*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

## Obsah

Poděkování	vi
Prohlášení	vii
Abstrakt	viii
Seznam zkratk	ix
<b>1 Úvod</b>	<b>1</b>
1.1 Motivace . . . . .	1
1.2 Cíle práce . . . . .	1
<b>2 Analýza</b>	<b>3</b>
2.1 Podporované rostliny . . . . .	3
2.2 Květináče - akademická řešení . . . . .	3
2.2.1 Hydroponická řešení . . . . .	4
2.2.2 Řešení klasické (používající substrát) . . . . .	7
2.3 Květináče - komerční řešení . . . . .	8
2.3.1 Výběr testovacího vzorku . . . . .	9
<b>3 Návrh a realizace měření</b>	<b>13</b>
3.1 Měřicí zařízení . . . . .	13
3.1.1 Hardware . . . . .	13
3.1.2 Software . . . . .	21
3.2 Webová aplikace . . . . .	21
3.2.1 Frontend . . . . .	22
3.2.2 Backend . . . . .	23
<b>4 Výsledky měření</b>	<b>27</b>
4.1 Relativní vlhkost vzduchu . . . . .	27
4.2 Teplota vzduchu . . . . .	28
4.3 Vlhkost půdy . . . . .	29
4.4 Intenzita světla . . . . .	30
4.5 Spotřeba elektrické energie . . . . .	32
<b>5 Testování</b>	<b>33</b>
5.1 Testování frontendu . . . . .	33
5.2 Testování backendu . . . . .	34
5.3 Testování senzorů a měřícího zařízení . . . . .	36
<b>6 Závěr</b>	<b>39</b>
Obsah přiloženého média	43

## Seznam obrázků

2.1	S. Bielikova - Easy grow (render) . . . . .	5
2.2	Schéma Windowfarm . . . . .	6
2.3	Nekonve (render) . . . . .	7
2.4	TREGREN T3 Kitchen Garden . . . . .	8
2.5	Véritable Potager Connect . . . . .	9
2.6	Produktový obrázek chytrého květináče Bentech Smart Garden . . . . .	10
2.7	Produktový obrázek chytrého květináče Tregren Genie . . . . .	11
3.1	Měřicí zařízení mezi dvěma testovanými vzorky. Na stole si můžeme kromě ohromného nepořádku všimnout přilepeného senzoru teploty a vlhkosti vzduchu, v každém květináči pak dalších senzorů. . . . .	14
3.2	Raspberry Pi Zero WH. . . . .	15
3.3	Grove Base Hat pro Raspberry Pi. . . . .	16
3.4	Grove light sensor v1.2. . . . .	17
3.5	Grove TemperatureHumidity Sensor (DHT11). . . . .	17
3.6	Grove Moisture Sensor. . . . .	18
3.7	Luxmetr GY-302 BH1750. . . . .	19
3.8	Chytrá zásuvka Shelly plug S . . . . .	19
3.9	Na této fotce je možno vidět zvolené zapojení senzorů na Grove Base Hat, který je přes GPIO připojen k pod ním se nacházejícímu Raspberry Pi. Modré šipky označují senzory patřící ke květináči Tregren, zelené patří k modelu od Bentechu. Červený je pak společný senzor teploty a vlhkosti vzduchu, zapojený do portu PWM. Ze čtveřice senzorů, zapojených do analogových portů (na obrázku vlevo) jsou dva horní senzory vlhkosti půdy, spodní dvojice jsou senzory intenzity světla. Přímou na prodloužené GPIO piny jsou pak připojeny luxmetry. Ty komunikují přes I2C sběrnici (I2C0, resp. I2C1) . . . . .	20
3.10	Schéma integrace knihovny Grove.py . . . . .	21
3.11	Schéma Symphony frameworku . . . . .	25
4.1	Graf relativní vlhkosti vzduchu v místě, kde se nacházely oba testované květináče, za měřené období (7. 4. 2021 – 28. 4. 2021) . . . . .	28
4.2	Graf teploty vzduchu v místě, kde se nacházely oba testované květináče, za měřené období (7. 4. 2021 – 28. 4. 2021) . . . . .	29
4.3	Graf vlhkosti půdy obou testovaných květináčů za měřené období (7. 4. 2021 – 28. 4. 2021) . . . . .	29
4.4	Graf intenzity světla obou testovaných květináčů za měřené období (7. 4. 2021 – 28. 4. 2021) . . . . .	30
4.5	Graf intenzity světla obou testovaných květináčů v luxech za měřené období (7. 4. 2021 – 28. 4. 2021) . . . . .	31
4.6	Graf spotřeby elektrické energie obou testovaných květináčů za měřené období (7. 4. 2021 – 28. 4. 2021) . . . . .	32

5.1	Okno se spuštěným testem pro frontend aplikace, výsledky a čas testování. Je možno si všimnout i hlášky, že Chrome je kontrolován automatizovaným testovacím softwarem. . . . .	34
5.2	Výstup testovací knihovny phpunit, bez chyb . . . . .	35
5.3	Výstup knihovny PHPStan nad zdrojovým kódem nám říká, že je nutno doplnit nativní typ vrácené hodnotě, dále ukazuje v jakém souboru a na jakém řádku. . . . .	35
5.4	Výstup knihovny PHP_CodeSniffer nad zdrojovým kódem nám tvrdí to samé, co PHPStan. Je proto vidět, že knihovny se občas i doplňují nebo dokonce překrývají. CodeSniffer nám navíc nabízí, že tento neduh na kódu umí automaticky opravit (poslední věta výstupu) . . . . .	36
5.5	Oba produkční vzorky a mezi nimi právě testované měřící zařízení. Zde je ještě vidět že nejsou zapojeny všechny senzory, stav je ještě před zasetím bazalky . . . . .	37

## Seznam tabulek

3.1	Hodnoty vlhkosti půdy . . . . .	18
-----	---------------------------------	----

## Seznam výpisů kódu

3.1	Požadavek na zásuvku s příkladem odpovědi . . . . .	20
-----	---	----

*Chtěl bych poděkovat především svojí rodině, která se mnou prožívala mnoho téměř kolapsových stavů při čekání na výsledky třetích pokusů zkoušek (a že jich bylo). Dále chci poděkovat vedoucímu této práce, Ing. Martinovi Daňhelovi, Ph.D. za veškerou pomoc a důležité rady a za ochotu mi pomáhat tuto práci dokončit. Nakonec bych rád poděkoval všem kantorům, kteří mě dostali až tam, kde jsem dnes. Díky vám všem.*



## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 27. června 2021

.....

## Abstrakt

Tato bakalářská práce se zabývá tématem chytrých květináčů. V práci jsou analyzovány podporované rostliny a akademická i na trh uvedená produkční řešení, která souvisí s automatizací nebo jakýmkoliv vylepšením pěstebního procesu rostlin. V analytické části je popsán výběr produkčních vzorků, které jsou vybrány za cílem testování a sběru produkčních dat během experimentálního pěstování. V praktické části je popsáno nejen sestavení měřicího zařízení, které sběr dat provádí, ale i architektura backendu a frontendu webové aplikace, kam jsou provozní data z měřicího zařízení zaznamenávána. Současně jsou v této části diskutovány a rozebrány naměřené hodnoty a na grafech, které jsou výsledkem frontendu webové aplikace, jsou porovnány hodnoty dvou různých produkčních květináčů. Hlavním přínosem této práce je vytvořený měřicí software a hardware, který se dá do budoucna použít jako základ pro prototyp fakultního chytrého květináče. Dalším přínosem jsou samotná data, naměřená po dobu experimentu, která vytváří měřítko pro budoucí testování fakultního prototypu. Na závěr je diskutována kvalita použitých produkčních vzorků a jsou rozvíjeny možnosti znovuvyužití měřicího zařízení. V příloze je pak možno nalézt zdrojové kódy aplikace a snímky z pěstitelského experimentu.

**Klíčová slova** chytrý květináč, monitorování zdrojů, automatické pěstování, Raspberry Pi, Symfony, Angular, Docker, REST

## Abstract

This bachelor thesis deals with the topic of smart flowerpots. The work analyzes supported plants and academic and market-based production solutions that are related to automation or some kind of improvement of the plant growing process. The analytical part describes the selection of production samples, which are selected for testing and collection of production data during experimental cultivation. The practical part describes not only the assembly of the measuring device, which performs data collection, but also the architecture of the backend and frontend of the web application, where the operational data from the measuring device is recorded. In the same part, the measured values are discussed and analyzed, and the values of two different production flowerpots are compared on the charts, which are a result of the frontend of the web application. The main benefit of this work is the created measuring software and hardware, which can be used in the future as a basis for a prototype of a faculty smart flower pot. Another benefit is the data itself, measured during the experiment, which determines the rate for future testing of the faculty prototype. Finally, the quality of the used production samples is discussed and the possibilities of reusing the measuring device are developed. In the appendix it is then possible to change the source codes of the application and images from the cultivation experiment.

**Keywords** smart flowerpot, sources monitoring, automated plant growth, Raspberry Pi, Symfony, Angular, Docker, REST

## Seznam zkratk

API	Application Programming Interface
DPS	Deska plošných spojů
GPIO	General-purpose input/output
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
JWT	JSON web token
LED	Light-Emitting Diode
MVC	Model View Controller
ORM	Objektově relační mapování
REST	Representational State Transfer
SPA	Single page application
SPA	Single Page Application



# Kapitola 1

## Úvod

Žijeme v době, kdy se tzv. chytré (smart) technologie rozšiřují do všech možných odvětví lidské činnosti. Lidé denně používají chytré telefony, chodí běhat nebo jezdit na kole s chytrými hodinkami na ruce a večer si třeba pouští filmy na chytré televizi. Smart technologie ovšem nezasáhly jen do sektoru zábavního nebo sportovního. Je možno se v domácnostech běžně setkat s chytrou lednicí, která je schopná si sama udělat diagnostiku nebo třeba oznámit majiteli aktuální teplotu uvnitř mrazáku pomocí mobilní aplikace. Není tedy velkým překvapením, že chytré už umí nějaký čas být i květináče. Ale co vlastně takový chytrý květináč umí? Kolik chytrý květináč stojí? A jak je asi složité nějaký podobný produkt postavit? Na tyto otázky se pokusí odpovědět tato práce.

### 1.1 Motivace

Klasické květináče jsou stále dominantním pomocníkem v domácím pěstování rostlin, ovšem čím dál více se začínají projevovat jejich slabiny. Doba zrychluje a klade na lidi daleko větší časové nároky, tudíž se často potýkají s nedostatkem času i na tak banální záležitosti jako zalít, pohnojit nebo jinak opečovávat jejich domácí květiny nebo byliny. A pokud je možno nahradit automatikou složité úkony, např. navádění dopravních letadel, není důvod, proč otálet s automatizací u jednoduchých činností jako pěstování rostlin. Přitom nejde o nijak ultra náročný úkol, a to z hlediska finančního, ani dovednostního. V době, kdy se dá relativně výkonný minipočítač pořídit za cenu dvou obědů v restauraci (Raspberry pi Zero se pohybuje na trhu kolem 350 Kč) a senzory v řádu desítek Kč, je finanční stránka o dost růžovější, než v minulosti. Navíc díky velmi dobře zpracovaným dokumentacím k mikrokontrolerům a minipočítačům, a také díky velkému zájmu elektro-kutilské komunity, se dnes prakticky kdokoli může během chvíle s těmito technologiemi seznámit, minimálně na uživatelské úrovni. Navíc pokud není motivací požitek z hrátek s elektronikou a sebezlepšení svých technických dovedností, může jí být stále nenasycený trh, kde se stabilně pohybuje pouze pár firem a jejichž výrobky jsou stále spíše okrajovou záležitostí. Chytré květináče stále ještě nejsou v domácnostech stejně samozřejmé stejně jako televize nebo počítač. Navíc prodávané kusy k tomu ani zcela nepřispívají svojí mnohdy dosti vysokou cenou. V segmentu chytrých květináčů je díky tomu stále prostor pro nové hráče, třeba s nějakým inovativním, neortodoxním nebo i, proti současné konkurenci, drtivě dostupnějším řešením.

### 1.2 Cíle práce

Tato práce se stane teoretickým základem pro budoucí snahu fakulty o vytvoření vlastního prototypu chytrého květináče, který by měl být předmětem některé návazné práce. V teoretické části

práce je rozebrána samotná problematika chytrého pěstování rostlin. Je zde diskutováno, jaké druhy rostlin jsou podporovány a jsou zde rozebrána již existující řešení, a to jak ze sféry akademické, tak i komerční. Existující produkční modely květináčů různých firem jsou srovnány podle rozdílných ukazatelů. Významnou roli u každého produktu hraje samozřejmě cena, ovšem ta není jediným důležitým parametrem při výběru produktu ke koupi. U chytrých květináčů se nabízí otázka, jak chytré opravdu dokáží být a na kolik svedou zastat práci a péči člověka, starajícího se o své rostlinky. V praktické části se práce zaměřuje na měření provozních ukazatelů u dvou zakoupených vzorků chytrých květináčů. Je zde popsán vybraný hardware (senzory, řídicí jednotka, jejich propojení) a software (skripty ovládající měřící zařízení, frontend a backend webové aplikace pro sběr dat, řešení serveru databáze apod.) a průběh komunikace měřícího hardwaru s webovou aplikací s uživatelským prostředím, ve které je možno naměřená data zobrazit na grafech a porovnávat je.

## Kapitola 2

# Analýza

Součástí zadání této práce je úkol analyzovat trendy v oblasti chytrých květináčů a podporovaných bylin. Bylo také potřeba, pro účel naměření provozních hodnot, vybrat nějaké již existující komerční vzorky.

### 2.1 Podporované rostliny

Druhů rostlin, které je možné pomocí chytrých květináčů pěstovat, je velké množství. Jedná se o různé bylinky, ovoce nebo zeleninu.

Například u výrobků od firmy Tregren, podporujících smart aplikaci pro mobilní telefony, prodejce uvádí databázi o velikosti 40 různých druhů [1], avšak realita je trochu jiná. SmartGardener, jak se aplikace jmenuje, na konci června 2021 ukazuje 10 druhů bylinek, mezi nimi třeba koriandr nebo tymián, 9 druhů salátů, 6 druhů ovoce nebo zeleniny a 1 květinu (orchidej)[2].

Další firma, která prodává chytré květináče, Click and Grow se může pochlubit databází o velikosti dokonce téměř 70 různých rostlin, samozřejmostí jsou všemožné bylinky jako majoránka, oregáno nebo pažitka, saláty a zelenina, jako u Tregrenu, najdeme tu však i více druhů okrasných květin.

Obě předchozí firmy ovšem drtivě válcuje (alespoň šíří databáze) produkt z Číny. Aplikace Flower Care, která je potřeba ke komunikaci s chytrým květináčem Xiaomi Smart Flowerpot, ukazovala v sekci „Plants“ řádově tisíce výsledků.

Skrze celý trh a různé značky se ovšem databáze chytrých aplikací i eshopy s připravenými pěstebními kapslemi nebo sadbovacími tabletami shodnou minimálně na jedné bylince - bazalce. Jde asi o nejpoblárnější rostlinu u výrobců chytrých květináčů, což je mimo jiné vidět i na produktových fotografiích, kde často právě bazalka dominuje.

Jako vhodný kandidát na testovací rostlinu byla tedy zvolena právě bazalka. Jednoletá, až 50 cm vysoká, velmi vonná bylina, která se pěstuje pro nať a listy, které se nejčastěji používají jako koření v kuchyni. Jedná se o rostlinu teplomilnou a vlhkomilnou a relativně nenáročnou na pěstování [3, 4]. Tomu, že se u bazalky jedná o vhodnou rostlinu pro pěstování v chytrých květináčích, svědčí i fakt, že je často přibalena ve formě semínek nebo dokonce předpřipravených pěstebních kapslí přímo s prodáváním modelem.

### 2.2 Květináče - akademická řešení

Do analýzy bylo vybráno také několik akademických řešení problému. Je pravda, že taková řešení mnohdy nabízejí jiný úhel pohledu oproti komerčním, jejichž hlavním cílem je masové rozšíření

a generování zisku. Z rešerše provedené v rámci systému digitální knihovny ČVUT Dspace<sup>1</sup> byly vybrány některé práce zabývající se taktéž tématem pěstování - bezúdržbového (za použití hydroponie) nebo chytrého (s využitím technologií a automatizace).

### 2.2.1 Hydroponická řešení

Tato řešení využívají principu hydroponického pěstování, které je založeno na absenci půdy a pěstování rostlin v živném roztoku. Pro tento druh pěstování je v podstatě zapotřebí jen nějaké médium, které slouží pro uchycení rostliny v květináči a živný roztok. Jako médium se používá například keramzit (umělé kamenivo), perlit nebo kokosové vlákno. Jde o materiály inertní, tudíž je potřeba do vody dostat živiny (organické a anorganické látky podporující růst), které jsou v klasické zemině přirozeně přítomny. K dodání živin se používá již zmíněný roztok.

Hydroponická řešení je možno rozdělit na pasivní a aktivní. Pasivní systémy jsou zcela nezávislé na elektrické energii a vystačí si jen s gravitací (tlak z vodní nádrže). Nejsou potřeba spínače a čerpadla. U takových řešení je pak nutný větší počet manuálních zásahů, jako dolévání vody, kontrola teploty, vlhkosti vzduchu, světla a dalších ukazatelů a přírodních vlivů. Díky technologické nenáročnosti a nulové spotřebě elektřiny jsou takové systémy zároveň ty s nižšími provozními náklady.

Druhá kategorie, systémy aktivní, se naopak vyznačuje použitím různých akčních členů, kupříkladu spínače pro hlídání intervalů zavlažování a přísunu živin. Při správném nastavení systému je pak možné se rostlinám nevěnovat třeba i několik dní v kuse.

Návrhem hydroponického květináče se zabývá bakalářská práce Stelly Bielikové s názvem *Hydroponický květináč*. [5] V této práci je představeno řešení v podobě chytrého hydroponického květináče se šesti otvory pro košíčky s inteligentní „půdou“. Řešení počítá i se signalizací nedostatku vody v nádržce. Ve spodní části se nachází pumpička, která má rozvířit vodu a pomoci s její cirkulací. Součástí květináče je i LED světlo, které co nejlépe imituje ideální světelné-tepelné podmínky, je teleskopické a osvětí celou plochu květináče. Ideální světelné podmínky mají za cíl maximalizovat fotosyntézu a ještě více zefektivnit celý proces růstu.

---

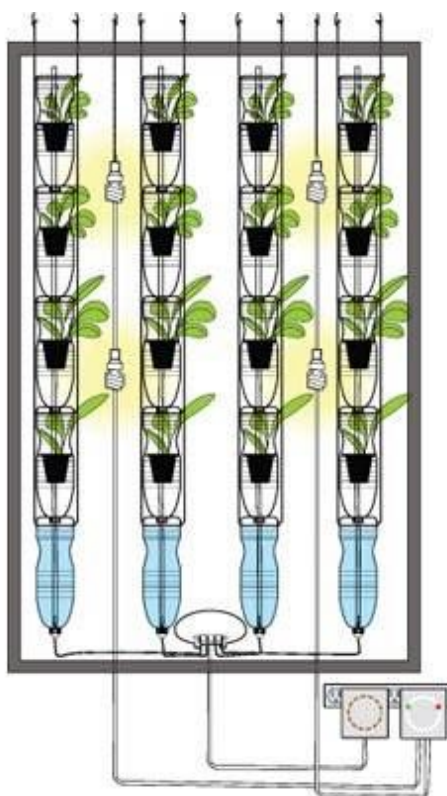
<sup>1</sup><https://dspace.cvut.cz/?locale-attribute=cs>





■ **Obrázek 2.1** S. Bielikova - Easy grow (render)

Jiný přístup k řešení zvolila Michaela Faktorová ve své diplomové práci s názvem *Nekonve*. [6] Pro tuto práci byla zvolena technika WindowFarm. WindowFarm umožňuje pěstování především plodin či bylinek, určených k přímé konzumaci a tím tedy snížení objemu dovážených a kupovaných surovin. Protože byl systém navrhnout především pro ty, kteří nemají možnost mít svoji vlastní zahrádku, a jediné místo, které obsahuje dost světla pro pěstování je okno, přešlo se na vertikální zavěšení jednotlivých nádob pod sebe právě pod oknem. Systémem cirkuluje voda pomocí hadiček. Ty vodu vedou z nádrže nad květináče, kterými následně roztok prokapává zpět k nádrži. Voda je do výšky vytahována s pomocí vzduchového čerpadla, které je tišší a spotřebovává méně energie než čerpadlo vodní. V jednotlivých nádobách jsou umístěny rostliny v hydroponických košíčcích naplněných keramzitem.



■ **Obrázek 2.2** Schéma Windowfarm

Řešení, diskutované v této práci, neuvažuje umělé osvětlení, neboť už z principu využívá přirozeného světla přicházejícího oknem. Dá se tedy odhadnout, že by mohlo být i energeticky méně náročné, ovšem za ne zcela slunných dnů by rostlina mohla trpět nedostatkem světla a tím pádem ve srovnání s předchozím řešením by růst nemusel být tak svižný. Navíc pokud je bráno jako faktor i roční období, poté je nutno konstatovat, že například v zimě by rostlina opravdu mohla mít velký nedostatek slunečního svitu (21. 12. 2020 byla doba od východu slunce po západ jen 8 hodin a 4 minuty) <sup>2</sup>

<sup>2</sup><https://www.meteogram.cz/vychod-zapad-slunce/ceska-republika/praha/>



■ Obrázek 2.3 Nekonve (render)

### 2.2.2 Řešení klasické (používající substrát)

Poslední citovanou prací z půdy ČVUT je bakalářská práce Vladimíra Čebíše s názvem *Automatický systém pro pěstování rostlin*. [7] Práce se zabývá alternativním řešením automatického systému pro pěstování zeleniny, ovoce či bylinek za pomoci autonomního systému v umělých podmínkách. Tento systém z dílny studenta fakulty elektrotechnické používá jako centrální jednotku mikrokontrolér programovatelný skrze vývojové prostředí Arduino. Toto řešení je nízkourovňové a bylo zvoleno autorem pro jeho schopnost být „sestaveno v co nejjednodušší, nejdostupnější a nejlevnější variantě.“ Autor použil dvě sebou navržené DPS v kooperaci se senzory a periferiemi (čerpadlo, tlačítka, LCD displej pro zobrazení hodnot). Práce je velmi zajímavá i z hlediska možnosti využití vytvořeného systému. Jak autor uvádí, „použití by mělo být univerzální, ať už do skleníku na chatě, nebo do nevyužitého prostoru v bytě“. Konečný vytvořený produkt pak opravdu velmi dobře funguje a z hlediska ceny je dosti konkurence schopný i oproti komerčním řešením.

Velmi podobným směrem se vydal i článek s názvem *Flower "Dock"- Single Chip Microcomputer-Based Smart Flowerpot*. [8] Jednoduchý systém pro monitorování pěstebních charakteristik rostliny je postaven na STC89C52, což je jednočipový mikropočítač, který provádí monitorování růstu v reálném čase tak, aby byly zachovány parametry prostředí, vhodné pro růst rostlin uvnitř květináče a které zaručují rostlinám stabilní a zdravé životní prostředí. Systém je navržen modulárně a přímo v článku je také ukázáno konkrétní využití s modulem snímáče teploty a vlhkosti půdy.

### 2.3 Květináče - komerční řešení

Článků a odborných prací se stejným nebo podobným tématem je více, ale řešení existuje mnoho už i v komerční sféře. Stačí zadat do nějakého vyhledávače heslo *Chytrý květináč* a objeví se záplava eshopů nabízejících tato řešení. Stejně jako v případě zmíněných akademických řešení i v komerční sféře lze najít oba hlavní druhy chytrých květináčů. Typy podporující hydroponní způsob pěstování i typy, kde se dá pěstovat klasicky v zemině. Na stránkách největšího českého eshopu Alza.cz je možno v kategorii chytré květináče <sup>3</sup> zobrazit nabídku sestávající se z různých řešení několika firem. Co se cenového rozpětí týče, jedná se (v době psaní práce) o výrobky s cenou od lehce více než dva tisíce korun až po důmyslná složitá řešení s cenovkou nad hranicí 7000 Kč.

Například nejlevnější chytrý květináč v nabídce alza.cz, TREGREN T3 Kitchen Garden [1], je malé hydroponické zařízení pro 6 sazenic. Výrobce uvádí, že je vyžadován speciální substrát, květináč disponuje samozavlažovacím systémem, speciálním růstovým světlem a údržba je potřeba minimální. Dále si zákazník může stáhnout aplikaci pro chytré telefony (platformy iOS / Android) a s její pomocí květináč ovládat. V aplikaci je na výběr ze 40 druhů rostlin, pro které je díky inteligentnímu systému vytvořeno ideální prostředí zajišťující jejich správný růst a prospívání. Aplikace dokáže zvolit správné nastavení LED světla, četnost zalévání a optimální doplňování živin. Dále je uvedeno, že díky vytvoření optimálního prostředí pro aktivní růst je ve srovnání s běžným pěstováním rostlin v zemině dosaženo až 3× vyšší výtěžnosti. Je vyžadováno připojení do elektřiny a komunikace s aplikací probíhá díky zabudovanému Bluetooth.



■ **Obrázek 2.4** TREGREN T3 Kitchen Garden

Na druhé straně cenového spektra stojí model s názvem Véritable Potager Connect. Ve výčtu

<sup>3</sup><https://www.alza.cz/chytre-kvetinace/18862180.htm>

jeho funkcí lze najít, že disponuje dvoulitrovou nádržkou na vodu, chytrým zavlažováním nebo třeba LED osvětlením s technologií Adapt'light „nahrazující slunce“. Pokud jde o podporu pro chytré telefony, je možno také stáhnout aplikaci, jak pro systém iOS, tak pro Android. Komunikace s telefonem potom probíhá stejně jako v případě levnějšího konkurenta, bezdrátově, přes rozhraní bluetooth. Do květináče se vkládá speciální substrát se semínky rostlin, v balení 4 ks substrátu se semínky bazalky, petržele, pažitky a tymiánu. Výrobce je uvedeno, že světlo 16 hodin svítí a 8 hodin je zhasnuté.



■ **Obrázek 2.5** Véritable Potager Connect

Minimálně z výrobce vypočteného seznamu funkcí se zdá být technika za dva tisíce korun a za sedm tisíc velmi podobná. Je tedy otázka, zda se vyplatí koupit jeden dražší model v hodnotě tří levnějších modelů.

### 2.3.1 Výběr testovacího vzorku

Pro potřeby práce bylo nutné vybrat testovací produkční vzorek, na kterém by bylo možné měřit provozní hodnoty při pěstování rostliny. V této práci bylo zvoleno pěstování v hlíně, proto bylo nutné zvolit vzorek, který podporuje pěstování tímto způsobem. Ještě ani toto zúžení výběru nebylo konečné, neboť chytré květináče umožňující tento typ pěstování se dále dělí na ty, kam se dá pouze umístit prefabrikovaná pěstební kapsle se speciálním substrátem a semínky, a na kusy podporující růst již zasazených květin. Pro co nejlepší přístup k rostlině a substrátu, ve kterém je zasazená, proto bylo vybíráno z řešení, do kterých lze umístit již existující rostlinu nebo kam se dá zasadit standardním způsobem.

Protože součástí této práce je i napsání webové aplikace, kde se budou nasbíraná provozní data zobrazovat, bylo zbytečné vybírat ze vzorků, kde již existují například aplikace pro chytrý

telefon nebo jiné formy zobrazení těchto údajů. Navíc díky absenci této rozšířené podpory bylo možné se s cenou dostat níže, než je tomu u produktů, které tyto vymoženosti poskytují takzvaně „out of the box“.

Aby se ovšem o vzorku dalo říct, že je to květináč chytrý, bylo nutné vybrat zařízení, vykazující alespoň nějaké znaky automatizace nebo jiného usnadnění procesu pěstování. Může se jednat třeba o automatické zapínání a vypínání světla v pravidelných intervalech nebo periodická cirkulace vody z nádržky pro osvěžení a zavlažení rostliny. Pokud jde o samotné umělé světlo, které například Windowfarm nevyužívá, bylo pro účel práce rozhodnuto o jeho nutnosti pro následný návrh fakultního květináče, který by měl být odolný i nepřízní světelných podmínek (kupříkladu při pěstování v suterénních bytech). Po průzkumu trhu skrz různé české i zahraniční eshopy byly nakonec vybrány dva produkční vzorky, které vykazovaly ideální parametry a jevily se jako schopní a nepříliš drazí reprezentanti chytrých květináčů.

### 2.3.1.1 Bentech Smart Garden

Produkt od firmy Bentech je 1,5 kg važící plastový květináč. Má dvě pěstební nádobky, nádržku až na 1,5 litru vody se senzorem, který má okamžitě upozornit při jejím nedostatku. Ke květináči je připojena LED lampa s otočnou hlavicí (360° horizontálně, 180° vertikálně), na které je možno si nastavit 3 režimy světla. LED lampa se skládá z 8 bílých, 8 červených a dvou modrých diod. Adaptér dodávaný s květináčem má vstupní napětí AC100 240V - 50/60HZ, výstupní DC12.8V/1A. Výrobcem udávaná spotřeba je 8,5 W.



■ **Obrázek 2.6** Produktový obrázek chytrého květináče Bentech Smart Garden

### 2.3.1.2 Tregren Genie

Jako druhý produkční vzorek byl vybrán chytrý květináč od již zmíněné firmy Tregren, což je přední finská společnost specializující se na vývoj a výrobu ekologických, moderních a praktických domácích minizahrádek využívajících technologii aktivního růstu. Cílem společnosti je přiblížit zdravé a chutné potraviny blíže ke spotřebiteli a to ve funkčním, čistém a elegantním provedení. Činnost společnosti byla zahájena již v roce 2010 ve Finsku a dnes je společnost jedním z předních zahradnických firem v Evropě<sup>4</sup>. Model Genie výrobce prezentuje jako „kuchyňskou zahrádku“. Skládá se z pěstební plošiny, pod kterou je nádržka na vodu s kapacitou až 1 litr. Ta je propojená s plošinou otvorem, kudy na plošinu v pravidelných intervalech čerpá vodu malé čerpadlo. Do základu zahrádky lze upevnit stojan s LED světlem. Adaptér má stejné vstupní a výstupní napětí jako předchozí květináč, udávaná spotřeba činí 5,5 W. Spotřeba ovšem nezahrnuje provoz čerpadla, to je poháněno čtyřmi AA bateriemi.



■ **Obrázek 2.7** Produktový obrázek chytrého květináče Tregren Genie

---

<sup>4</sup><https://www.tregren.com/page/15/>





# Návrh a realizace měření

## 3.1 Měřící zařízení

Po výběru vhodných produkčních vzorků přišel na řadu úkol sestavit měřící zařízení, které měří a zaznamenává provozní hodnoty při samotném testovacím pěstování. Měřící zařízení se skládá z několika kusů hardware, řídicího počítače, senzorů a doplňkových měřících zařízení. Sestavu pak doplňuje software, většinou psaný v pythonu [9], který hardware ovládá a posílá naměřená data na backend webové aplikace.

### 3.1.1 Hardware

Hardware měřícího zařízení se sestává z jednoho mini počítače, ve funkci řídicí jednotky, několika senzorů a chytré zásuvky. Sensory jsem pořídil ve dvou kusech (pro každý květináč jeden), až na senzor teploty a vlhkosti vzduchu, který byl, díky uložení obou květináčů do jedné místnosti, potřeba jen jeden.

#### 3.1.1.1 Raspberry Pi Zero WH

Jako mozek celé měřící soustavy byl zvolen minipočítač od nadace Raspberry Pi Foundation [10]. Jde o jejich nejmenší model počítače s rozměry pouhých 66.0mm x 30.5mm x 5.0mm, tudíž vhodný jako prostorově nenáročná řídicí jednotka. Raspberry Pi Zero je osazen jednojádrovým procesorem ARMv6 taktovaným na frekvenci 1GHz, grafickým procesorem VideoCore IV GPU a pamětí s kapacitou 512 MB. K dispozici je port micro USB pro napájení a druhý micro USB se specifikací OTG. Tento minipočítač lze zakoupit ve čtyřech specifikacích:

- Zero - základní model
- Zero W - model s integrovanou WiFi 802.11n a Bluetooth 4.0 konektivitou
- Zero H - model osazený GPIO headerem
- Zero WH - kombinace vlastností modelu W a H

Model s osazením GPIO headerem byl zvolen z důvodu připojení senzorů k počítači. Bezdrátová konektivita byla vyžadována pro komunikaci s chytrými zásuvkami měřícími spotřebu a pro komunikaci počítače se serverem, na kterém běží backend webové aplikace a kam Zero posílá naměřená data. Samotný počítač ovšem nedisponuje žádnou permanentní pamětí. Systém a data jsou uloženy na microSD kartě, pro kterou je na Zero dedikovaný slot.



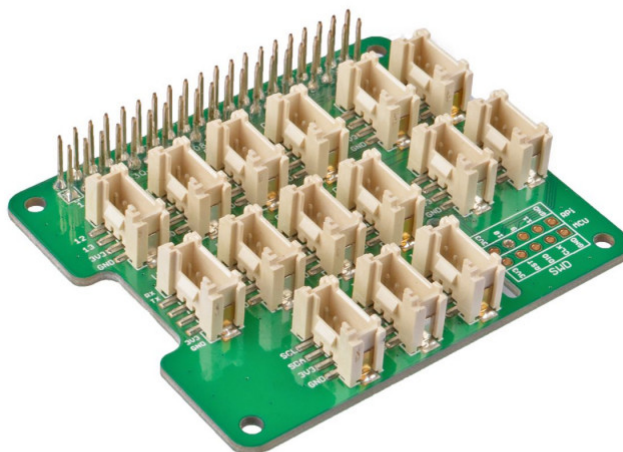
■ **Obrázek 3.1** Měřící zařízení mezi dvěma testovanými vzorky. Na stole si můžeme kromě ohromného nepořádku všimnout přilepeného senzoru teploty a vlhkosti vzduchu, v každém květináči pak dalších senzorů.



■ Obrázek 3.2 Raspberry Pi Zero WH.

### 3.1.1.2 Grove Base Hat pro Raspberry Pi

Jako druhá komponenta byl vybrán Grove Base Hat od firmy Seeed. Grove je modulární standardizovaný konektorový systém pro prototypování. Pro standard Grove lze zakoupit různé vstupní i výstupní zařízení (senzory, tlačítka apod.), které lze pomocí 4 pinového konektoru připojit na Grove Hat. Na tomto hatu, který se k Raspberry Pi připojí díky 40 GPIO pinům se nachází digitální, analogové, I2C, PWM a UART porty. Pro měřicí zařízení v této práci bylo využito připojení 4 analogových portů (senzory vlhkosti půdy a světla) a jednoho PWM pro připojení senzoru vlhkosti a teploty vzduchu. Výhodou je také protažení GPIO pinů, na které se dá dále připojit i senzor nebo periferie nekompatibilní se standardem Grove.

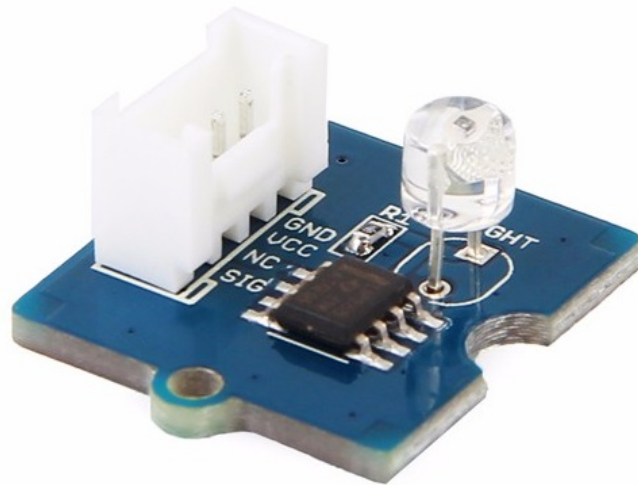


■ **Obrázek 3.3** Grove Base Hat pro Raspberry Pi.

### 3.1.1.3 Grove senzor světla

Pro měření světla byl původně zvolen senzor Grove light sensor v1.2, který integruje fotorezistor pro detekci intenzity světla, kdy odpor se snižuje pokud se zvyšuje intenzita světla. Na senzoru lze najít čip LM358 s operačním zesilovačem, který produkuje napětí korespondující s intenzitou světla. Výstupem je analogová hodnota - vyšší hodnota značí jasnější světlo a naopak. Pro připojení ke Grove Base Hat je možno využít analogové porty A0, A2, A4 nebo A6.

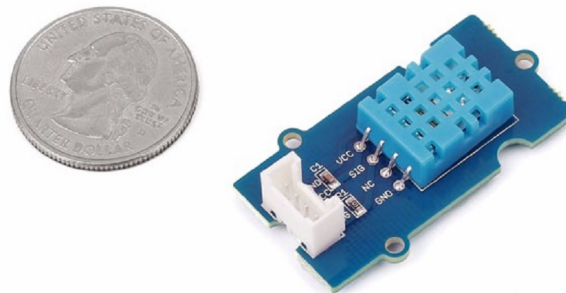
Bohužel během testování hardware bylo zjištěno, že senzor světla má úzké rozmezí hodnot. I slabé světlo stoupalo v naměřených hodnotách k maximu a intenzita se tedy nedala u silnějšího světla efektivně rozlišit. Proto bylo nutné sáhnout pro řešení zmíněné v sekci níže.



■ **Obrázek 3.4** Grove light sensor v1.2.

#### 3.1.1.4 Grove senzor teploty a vlhkosti vzduchu

Teplota a vlhkost vzduchu je měřena za pomoci senzoru DHT11, také z rodiny Grove. Senzor integruje kapacitní senzor pro měření relativní vlhkosti (20-90%). Teplotu v okolí snímá negistor (termistor s negativním teplotním koeficientem, se zahřátím klesá odpor), v rozpětí 0-50 °C. Pro připojení ke Grove Base Hat je možno využít jakéhokoliv GPIO pinu.



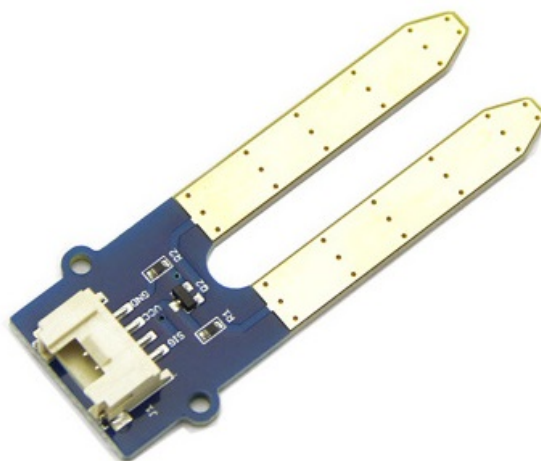
■ **Obrázek 3.5** Grove TemperatureHumidity Sensor (DHT11).

#### 3.1.1.5 Grove senzor vlhkosti půdy

Měření vlhkosti půdy zajišťuje Grove Moisture Sensor, který měří elektrický odpor mezi elektrodami. Hodnoty ze senzoru se dají podle dokumentace aplikovat na stav půdy podle tabulky níže:

■ **Tabulka 3.1** Hodnoty vlhkosti půdy

Hodnota	Význam
0 - 300	Suchá půda
300 - 700	Mokrá půda
700 - 950	Voda

■ **Obrázek 3.6** Grove Moisture Sensor.

### 3.1.1.6 Luxmetr GY-302 BH1750

Po zjištění, že senzor pro měření světla z rodiny Grove nemusí dostatečně vyhovovat rozsahem pro účely měření a rozlišení výsledků, bylo nutné nalézt doplňující řešení, proto byl zvolen tento měřič, který poskytuje hodnotu světla v jednotce lx s měřeným rozmezím 0 - 65535 lux. Luxmetr je osazen čipem BH1750FVI. Pro připojení používá měřič sériovou sběrnici I<sup>2</sup>C. Pro připojení k měřicímu zařízení byl tedy využit prodloužený GPIO header na Grove base hat.



■ **Obrázek 3.7** Luxmetr GY-302 BH1750.

### 3.1.1.7 Shelly plug S

Pro měření spotřeby elektrické energie byla zvolena chytrá zásuvka od firmy Shelly. Tato zásuvka integruje Wi-Fi modul a umí se chovat jak jako přístupový bod, tak i jako klient. Pro potřeby měřicího zařízení byla využita možnost zásuvku nastavit jako klienta a připojit jí k přístupovému bodu, který vytvořilo Zero. Velice užitečnou funkcí je také existence HTTP API zásuvky, přes které je možno zásuvku kontrolovat (zapínat/vypínat) a také odečítat údaje o provozu. Na webových stránkách výrobce je API velmi dobře zdokumentováno <sup>1</sup>.



■ **Obrázek 3.8** Chytrá zásuvka Shelly plug S

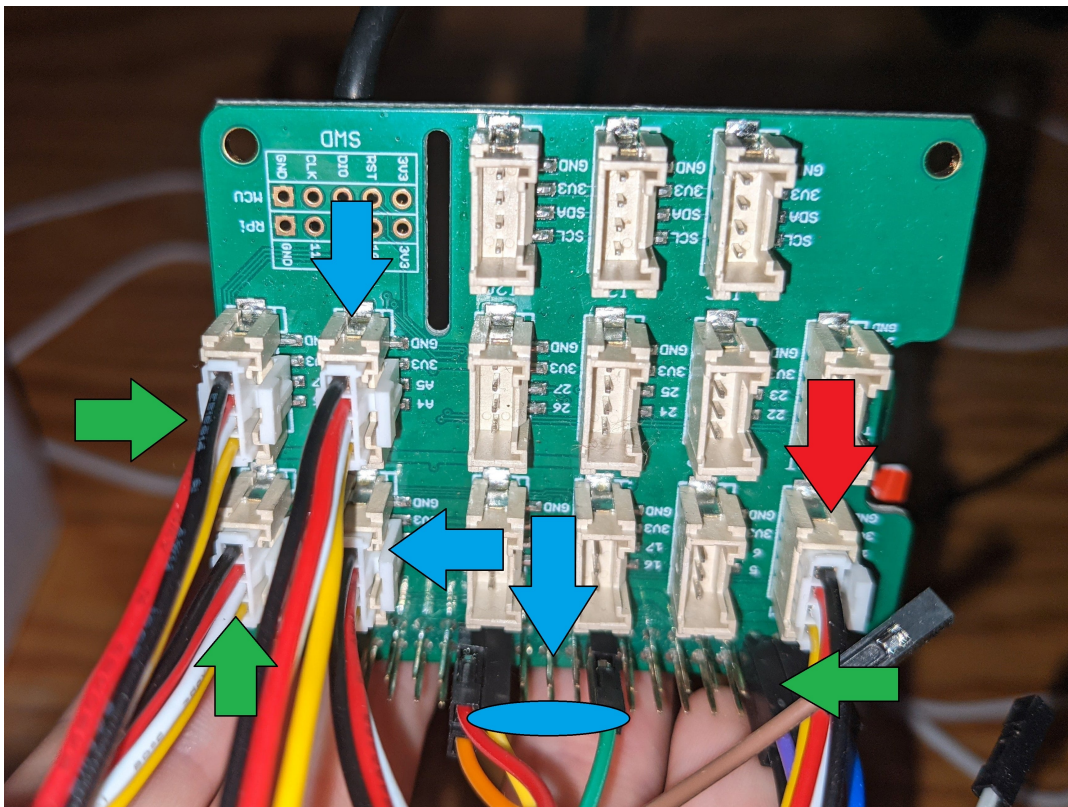
<sup>1</sup><https://shelly-api-docs.shelly.cloud/shelly-plug-plugs>

■ **Výpis kódu 3.1** Požadavek na zásuvku s příkladem odpovědi

```
GET /meter/0

{
  "power": 0,
  "overpower": 23.78,
  "is_valid": true,
  "timestamp": 0,
  "counters": [1,2,3], //Watt-minute value for last 3 minutes
  "total": 4
}
```

### 3.1.1.8 Detail zapojení senzorů



■ **Obrázek 3.9** Na této fotce je možno vidět zvolené zapojení senzorů na Grove Base Hat, který je přes GPIO připojen k pod ním se nacházejícímu Raspberry Pi. Modré šipky označují senzory patřící ke květináči Tregren, zelené patří k modelu od Bentechu. Červený je pak společný senzor teploty a vlhkosti vzduchu, zapojený do portu PWM. Ze čtveřice senzorů, zapojených do analogových portů (na obrázku vlevo) jsou dva horní senzory vlhkosti půdy, spodní dvojice jsou senzory intenzity světla. Přímou na prodloužené GPIO piny jsou pak připojeny luxmetry. Ty komunikují přes I2C sběrnici (I2C0, resp. I2C1)



### 3.1.2 Software

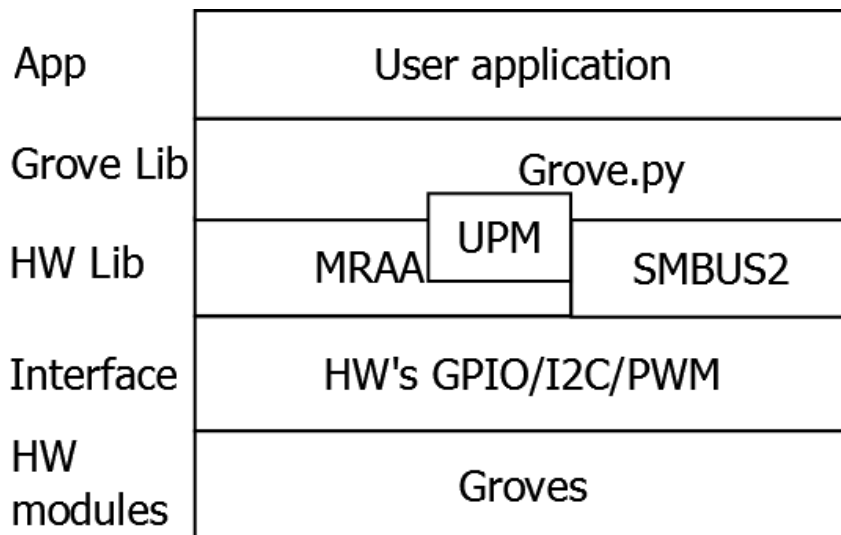
Ovládací software zařízení se skládá z operačního systému řídicího mini počítače, knihovny pro čtení dat ze senzorů rodiny Grove a skriptů v pythonu, které to vše zapouzdřují do pár tříd (a jednoho Git repozitáře). Měření hodnot probíhá každou minutu, jednotlivé skripty jsou volané cronem <sup>2</sup>. Hodnoty se po naměření odesílají jako HTTP požadavky na vzdálený server, kde běží v další sekci popsaná webová aplikace.

#### 3.1.2.1 Raspberry Pi OS

Jako operační systém byl pro Raspberry Pi Zero zvolen doporučený a výrobcem podporovaný Raspberry Pi OS <sup>3</sup>, který je kompatibilní se všemi počítači Raspberry Pi (ne však pro mikrokontrolér Raspberry Pi Pico). Raspberry Pi OS je unixový systém založený na debianu, je vysoce optimalizovaný pro práci na jednodeskových počítačích s ARM procesorem <sup>4</sup>.

#### 3.1.2.2 Grove.py

Grove.py je knihovna napsaná v Pythonu, speciálně vytvořená pro podporu práce s Grove moduly. Je optimalizovaná pro práci s vestavnými systémy, speciálně pro Coral Dev Board, NVIDIA Jetson Nano a Raspberry Pi <sup>5</sup>. K provozu senzorů grove závisí grove.py na mnoha knihovnách hardwarového rozhraní, jako je mraa / upm / smbus2.



■ Obrázek 3.10 Schéma integrace knihovny Grove.py

## 3.2 Webová aplikace

Ukládání měřených provozních dat a následnou možnost náhledu na ně zprostředkovává webová aplikace. Aplikace je v uživatelské části koncipovaná jako takzvaná SPA (Single Page Application), což znamená, že po prvotním načtení už prohlížeč neobnovuje stránku a všechny síťové

<sup>2</sup>Cron je softwarový démon, který v operačních systémech automatizovaně v konfigurovatelných periodách spouští v nějaký příkaz resp. proces

<sup>3</sup><https://www.raspberrypi.org/software/operating-systems>

<sup>4</sup><https://www.arm.com/why-arm/architecture>

<sup>5</sup><https://github.com/Seeed-Studio/grove.py>

požadavky jsou posílány na pozadí aplikace a zpracovány javascriptem. Proto je rozdělena na dva spolu komunikující nezávislé díly:

1. Frontend - Grafická webová aplikace postavená na javascriptovém frameworku
2. Backend - Kontejnerované REST api postavené na PHP frameworku

Obě části mezi sebou komunikují pomocí HTTP požadavků a odpovědí ve formátu JSON.

### 3.2.1 Frontend

Pro zobrazení naměřených provozních dat byla vytvořena grafická webová aplikace, založená na javascript frameworku Angular. Jedná se o jeden ze tří nepoužívanějších javascript frontend frameworků [11], o jehož vývoj se stará hlavně Google. Vedle Angularu bylo uvažováno o dalších dvou, velmi často užívaných frameworkcích, Reactu (většinou se o vývoj stará Facebook) a Vue.js (komunitní projekt, avšak s velkým dosahem, používán například Githubem). Angular byl nakonec vybrán z této trojice hlavně pro předešlé zkušenosti s psáním aplikací s jeho pomocí.

Díky použití javascript frameworku je frontend aplikace pojata jako SPA. Pokud jde o samotný javascript, zdrojový kód je psaný v TypeScriptu. TypeScript je open-source programovací jazyk vytvořený a spravovaný firmou Microsoft. Jedná se o nadstavbu nad jazykem javascript, která jej rozšiřuje o statické typování a další atributy, které známe z objektově orientovaného programování. Samotný kód psaný v TypeScriptu se pak kompiluje do JavaScriptu. [12]

Samotné posílání požadavků mezi frontendem a backendem aplikace probíhá pomocí JSON zpráv, autentizace probíhá prvotně zadáním uživatelského jména a hesla, po získání tokenu JWT (JSON web token) se každý další požadavek autentizuje už jen pomocí něj. JWT je otevřeným průmyslovým standardem (RFC 7519) pro bezpečné zastupování nároků mezi dvěma stranami. JSON Web Token (JWT) je kompaktní způsob reprezentace bezpečný pro URL. Data jsou v JWT zakódována jako objekt JSON, který se používá jako payload JSON webového podpisu (JWS) nebo jako prostý text JSON web šifrování (JWE), která umožňuje digitální zpracování deklarací podepsaných nebo integritně chráněných pomocí ověřovacího kódu zprávy (MAC) a / nebo šifrovaných. Aplikace je rozdělena do komponent:

- HomeComponent se stará o vykreslení denních grafů všech sledovaných provozních ukazatelů, pomocí utility kalendář je možné zvolit datum
- LongtermComponent zajišťuje vykreslení grafu jednoho zvoleného ukazatele, avšak s různě dlouhým časovým rozsahem, pomocí dvou instancí utility kalendáře je možné zvolit data od a do
- LoginComponent obstarává přihlašovací formulář a autentizaci aplikace

Každá komponenta v aplikaci má dvě složky, html šablonu a kontroler (Typescript třída). Kontroler interaguje s šablonou prostřednictvím API metod a property-bindingu. Poněvadž se jedná o Single page aplikaci, klasický routing s použitím proklikávání odkazů musí být řešen také javascriptem. S tím Angularu pomáhá modul Router. Pro zobrazení výsledků v grafech byla použita velmi populární javascriptová open-source knihovna Chart.js. Knihovna používá HTML5 canvas, umí zobrazit až 8 druhů grafů a poradí si i s responzivním překreslováním grafů, což je v době širokého používání zařízení s menším displejem nutnost. Samotná aplikace je taktéž stylována pro všechny velikosti viewportu zařízení, díky použití Bootstrap CSS frameworku, který je, jak se na jeho stránkách uvádí, „mobile-first“.

## 3.2.2 Backend

Backend webové aplikace je postavený na Php frameworku Symfony<sup>6</sup>. Jako databáze byl zvolen stroj MySQL, webový server obstarává Nginx. S kontejnerizací aplikace včetně databáze pomáhá software Docker. Celá tato soustava běží na VPS s operačním systémem Ubuntu minimal 20.04 (pro kontejnerové aplikace je minimální distribuce vhodná, popřípadě ještě lepší je OS přímo vytvořený pro provoz kontejnerů).

Takzvaný LEMP stack (Linux, Engine-x, MySQL / MariaDB, Php) byl vybrán hlavně kvůli nízkým provozním nákladům, neboť jde vesměs o open-source řešení, bez nutnosti platit licenční poplatky. Zároveň jde o velice často používané technologie, například jazyk Php je používaný necelými 80 procenty všech světových webů<sup>7</sup>. I díky těmto číslům je kolem těchto technologií široká komunita a velký výběr knihoven a frameworků, pomáhajících s vývojem.

### 3.2.2.1 Docker

Docker Engine<sup>8</sup> je runtime kontejneru, který běží na různých operačních systémech Linux (CentOS, Debian, Fedora, Oracle Linux, RHEL, SUSE a Ubuntu) a Windows Server. Docker vytváří jednoduché nástroje a univerzální přístup k balení, který sdružuje všechny závislosti aplikací uvnitř kontejneru, který se poté spouští na Docker Engine. Docker Engine umožňuje kontejnerovým aplikacím konzistentně běžet kdekoli na jakékoli infrastruktuře, řeší závislosti za vývojáře a operační týmy a eliminuje problém „na mém notebooku to funguje“. Velkou výhodou a důvodem, proč byl Docker použit v této práci, je velmi snadná přenositelnost aplikací mezi různými fyzickými stroji (například v případě nutnosti změnit poskytovatele serveru). S pomocí speciálního Makefile bylo dosaženo stavu, kdy je celý kontejner s backend částí aplikace spustitelný na jeden příkaz, hned po naklonování nejnovější verze z git repozitáře. Odpadá tudíž potřeba v případě migrace na nový server instalovat manuálně všechny závislosti, kofingurovat databázi, server a ostatní.

### 3.2.2.2 MySQL

Jako databázový stroj byl vybrán MySQL, vlastněný firmou Oracle. Podle údaje na vlastních webových stránkách jde o „nejpopulárnější open-source databáze“. Při pohledu do různých reportů podílů na trhu [13], opravdu se umísťuje na předních příčkách v popularitě a používanosti (spolu s PostgreSQL). Jde o relační databázi, používanou pro persistenci dat aplikace. MySQL je multiplatformní databáze. Komunikace s ní probíhá – jak už název napovídá – pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními. V aplikaci je použita nejnovější podverze MySQL 8. Zvažována byla v rámci práce i ostatní open-source řešení, již zmiňovaný PostgreSQL nebo MariaDB (což je komunitou vyvíjená nástupnická větev MySQL). Ovšem zkušenost s autorem s MySQL nakonec rozhodla.

Vedle tradičních relačních databází existuje i NoSQL typ, ovšem ten v tomto případě není vhodnou volbou z důvodu realční povahy ukládaných dat. Databáze jako MongoDB by našla využití například v případě rozšíření aplikace o logování chybových stavů a jejich popisu.

### 3.2.2.3 Nginx

Ve světě web serverů existují dva velcí globální hráči[14, 15]:

- Apache - vytvořen 1995, předinstalován v mnoha Linux distribucích
- Nginx - vytvořen 2004, prvotně hlavně pro zrychlení načítání statických stránek, později přidána podpora dynamického obsahu

<sup>6</sup><https://symfony.com/what-is-symfony>

<sup>7</sup>[https://w3techs.com/technologies/overview/programming\\_language](https://w3techs.com/technologies/overview/programming_language)

<sup>8</sup><https://www.docker.com/products/container-runtime>

Pro tuto práci byl zvolen Nginx. Jde o webový server, který se více hodí pro projekty s omezenými systémovými prostředky, není tak robustní a náročný jako Apache a vykazuje i rychlejší odezvu v mnoha scénářích. Nginx používá asynchronní, neblokující architekturu založenou na událostech, narozdíl od Apache serveru, který pro každý požadavek vytváří nový proces. Je to dáno i možností serveru Apache rekurzivně konfigurovat pro každý adresář server jinak pomocí `.htaccess` souborů<sup>9</sup>. Proto se Apache hodí například pro projekty sdílených hostingů, ale pro menší projekty bez potřeby konfiguraci rozsekávat je výhodnější použít Nginx.

#### 3.2.2.4 Php a Symfony framework

Poslední a velmi důležitou součástí je MVC Php framework Symfony. Jedná se o jeden z nejpoužívanějších a nejpoužívanějších Php frameworků [16], tudíž se vývojáři mohou spolehnout na velkou podporu komunity při řešení problémů. Symfony obsahuje velký počet komponent, které usnadňují vývoj, například:

- Security - poskytuje infrastrukturu pro sofistikované autorizační systémy
- Routing - zajišťuje mapování HTTP požadavků na kontrolery a akce
- Console - usnadňuje tvorbu konzolových příkazů a skriptů
- Filesystem - zapouzdřuje základní utility pro práci se souborovým systémem

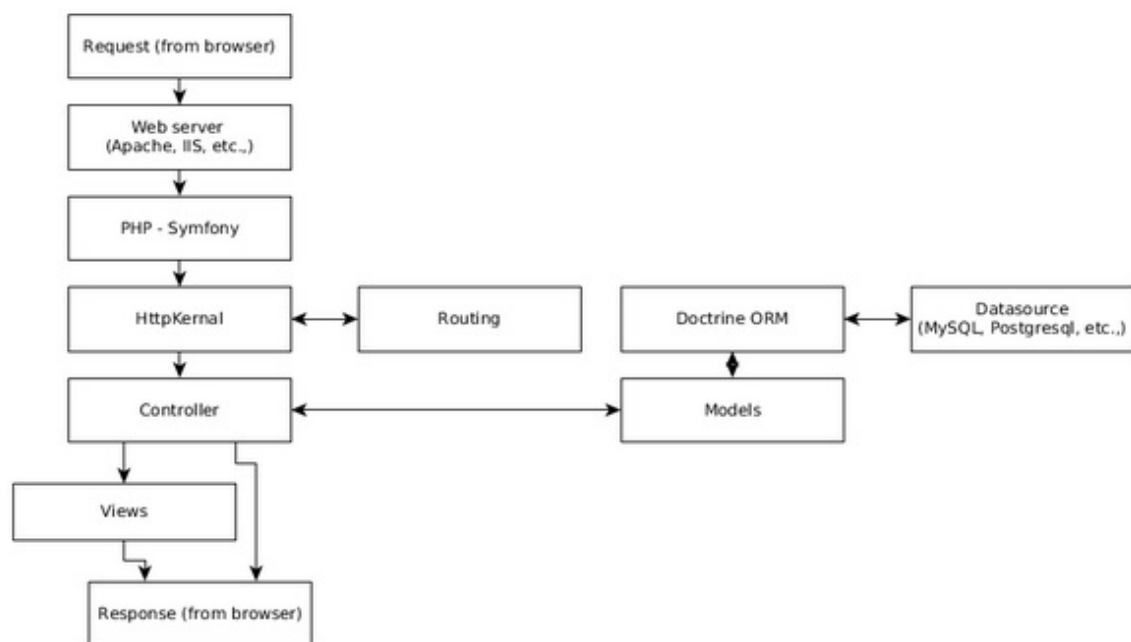
Pro komunikaci s databází je pak použita ORM knihovna Doctrine, která je velmi jednoduše do Symfony integrovatelná. Doctrine pracuje s databází pomocí entit, kdy každá entita v ORM reprezentuje tabulku v databázi. Doctrine samozřejmě řeší i důležité bezpečnostní aspekty propojení aplikace s databází, jako třeba ochranu SQL injection.

Symfony je taky velmi praktický pomocník pro práci na dlouhodobých projektech, neboť nabízí kromě nejnovějších verzí i takzvanou *long-term support* verzi, jejíž výhoda je v daleko delší době podpory než jakou mají standardní verze. LTS je podporována vývojáři po dobu 4 let, narozdíl od nejnovější stabilní verze, která má podporu vždy jen 6 měsíců. [17]

Pro komunikaci Symfony aplikace s Raspberry Pi Zero i pro komunikaci s frontendem byl zvolen standard JWT. Samotná aplikace následuje koncept DDD (Domain driven design). Služby, kontrolery, entity a další třídy jsou rozdělené do BC (Bounded context):

- Core - zde se nachází třídy používané skrz celou aplikaci
- Data - obsahuje aplikační kontrolery vystavující endpointy pro získávání dat frontendem
- Flowerpot - nacházejí se zde třídy související s entitou květináče, například konzolový příkaz pro vytvoření nového květináče v databázi nebo repozitář pro zobrazení existujících květináčů
- Log - zde najdeme třídy související s entitou logu a typu logů (vlhkost půdy, světelná intenzita apod.), taktéž služby pro vytváření nových logů a aplikační kontroler, který vystavuje endpoint pro samotné logování. Je zde i třída popisující strukturu dat, která jsou posílána na frontend nebo příkaz pro přidání typu logů
- Ping - s pomocí kontrolerů v této složce lze ověřit samotnou funkčnost aplikace, aniž by bylo potřeba rovnou začínat s logováním, obsahuje endpoint jak pro neautentifikovanou, tak i pro autentifikovanou odpověď (ping)
- User - zde se nachází všechny třídy a služby související s managementem uživatelů, jakožto i konzolový příkaz pro vytvoření uživatele nového (důležité při čisté instalaci aplikace)

<sup>9</sup><https://www.nginx.com/resources/wiki/start/topics/examples/likeapache-htaccess/>



■ Obrázek 3.11 Schéma Symfony frameworku



# Výsledky měření

Po zasetí bazalky a zdárném sestavení, zapojení a testování měřicího zařízení, které trvalo trochu déle, než bylo očekáváno, se konečně mohlo přejít ke sbírání dat o pěstování. Obdobím, ze kterého jsou níže komentovány výsledky, jsou tři týdny v dubnu 2021, konkrétně 7. dubna - 28. dubna 2021. Pro toto období byla sbírána data ze čtyř senzorů:

- Senzor teploty a vlhkosti vzduchu
- Senzor vlhkosti půdy
- Senzor intenzity světla
- Luxmetr

Senzor teploty a vlhkosti vzduchu je pro oba květináče společný, neboť byly umístěny vedle sebe a tudíž měly oba modely totožné klimatické podmínky. Sestavu doplňuje chytrá zásuvka, která bezdrátově odesílala do měřicího zařízení data o odběru elektrické energie. Níže komentované grafy jsou screenshoty z frontend aplikace.

### 4.1 Relativní vlhkost vzduchu

Prvním komentovaným ukazatelem je relativní vlhkost vzduchu. Jedná se o veličinu udávanou v procentech, vyjadřující jak moc se množství vodních par ve vzduchu blíží bodu nasycení. Důležité je také zmínit, že relativní vlhkost je různá pro rozdílné teploty. 1 metr krychlový vzduchu, mající relativní vlhkost 50 %, obsahuje při [18]:

- 30°C cca 15,02 g H<sub>2</sub>O
- 20°C cca 8,6 g H<sub>2</sub>O
- 10°C cca 4,66 g H<sub>2</sub>O
- 5°C cca 3,8g H<sub>2</sub>O

V grafu můžeme během měřicího období vidět oscilaci relativní vlhkosti s minimální hodnotou 17 % a maximální 38 %. Zajímavé je porovnání hodnot s počasím v době experimentu. Například maximální hodnota ze dne 12. dubna 2021 docela jasně koreluje s venkovním podnebím, kdy venku bylo deštivo a snížila se teplota, což obojí patří mezi faktory pro zvýšení relativní vlhkosti vzduchu. Naopak v závěru měření 27. dubna se relativní vlhkost dostala téměř ke svému minimu, což opět souhlasí s venkovním vývojem, kdy týden v kuse srážkový úhrn ukazoval nulu a teploty



■ **Obrázek 4.1** Graf relativní vlhkosti vzduchu v místě, kde se nacházely oba testované květináče, za měřené období (7. 4. 2021 – 28. 4. 2021)

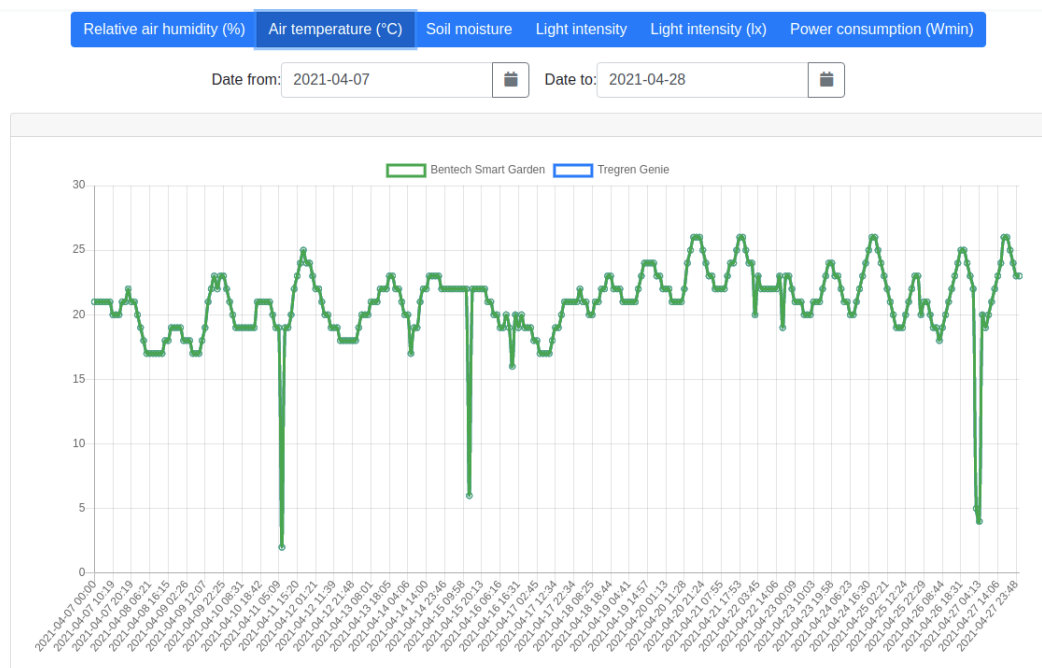
se pomalu šplhaly výše. [19] Celkově, za celé období měření, byla relativní vlhkost vzduchu v pokoji s květináči spíše nižší než by měla být, vzhledem k vlhkomilnosti bazalky. Naměřené hodnoty docela dobře korelují s průběhem počasí a proto je možno říct, že senzor měřil dostatečně kvalitně.

## 4.2 Teplota vzduchu

Pokud jde o záznam teploty v místě pěstování, můžeme vidět stabilní oscilaci mezi 17 až 26 stupni Celsia. Ovšem je možné si na grafu všimnout i občasných výkyvů k hodnotě 5 stupňů. Prozkoumání sousedních hodnot nám napoví, že výkyv opravdu není dán náhlým promrznutím pokoje, ve kterém testování probíhalo, spíše jde o vadné hodnoty, které by bylo třeba z produkčního řešení vyfiltrovat, popřípadě iniciovat v případě naměření tak rozdílné hodnoty nové měření ve stejné periodě. U těchto vadně naměřených hodnot se zpravidla nestává, že by se objevily i jen dvakrát za sebou, tudíž jako řešení se nabízí druhé a třetí kontrolní měření. Jako algoritmická detekce takto vadné hodnoty, neboť počítače nedisponují lidskou intuicí, se nabízí použití srovnání naměřené hodnoty s klouzavým průměrem za posledních N period. Pokud se hodnota liší o více než je nastavená určitá mez, spustí se kontrolní přeměření. Senzor, který měří teplotu je totožný s tím, který měří relativní vlhkost, proto je zvláštní, že u teploty vykazuje občasně nekonzistence. Bylo by možné situaci také řešit pořízením jiného, přesnějšího senzoru, určeného pouze pro tento účel.

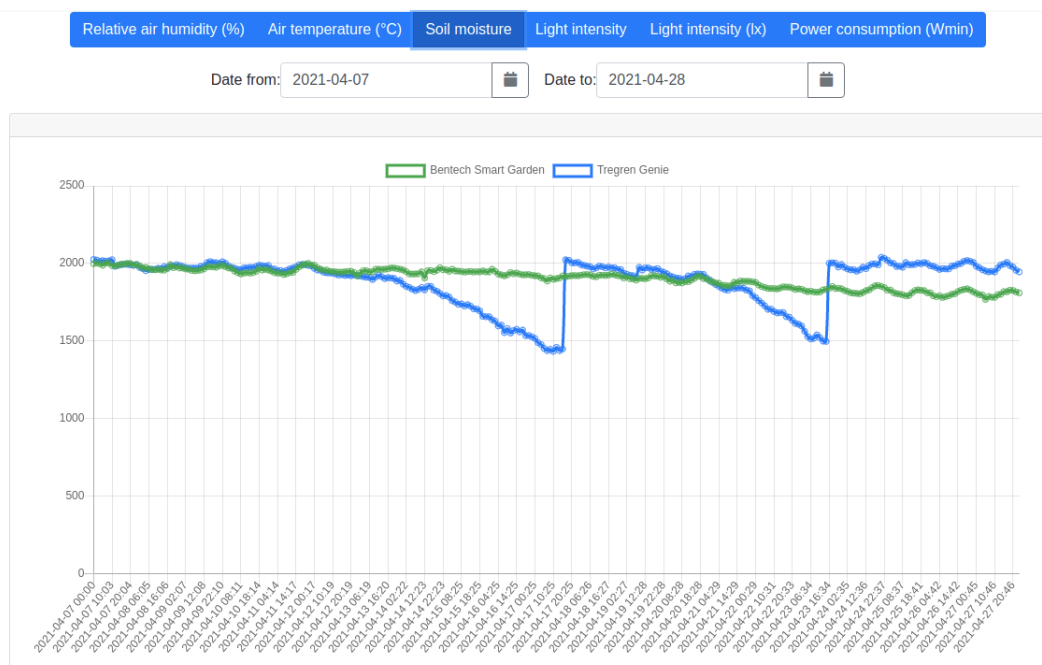
Naměřené hodnoty teplot ovšem stále nejsou optimální pro rostlinu, jakou je bazalka. Výkyvy teploty i relativní vlhkosti se samozřejmě dají minimalizovat s použitím skleníkové metody pěstování, nebo jiné metody s kontrolovaným vzduchovým prostředím.





■ **Obrázek 4.2** Graf teploty vzduchu v místě, kde se nacházely oba testované květináče, za měřené období (7. 4. 2021 – 28. 4. 2021)

### 4.3 Vlhkost půdy



■ **Obrázek 4.3** Graf vlhkosti půdy obou testovaných květináčů za měřené období (7. 4. 2021 – 28. 4. 2021)

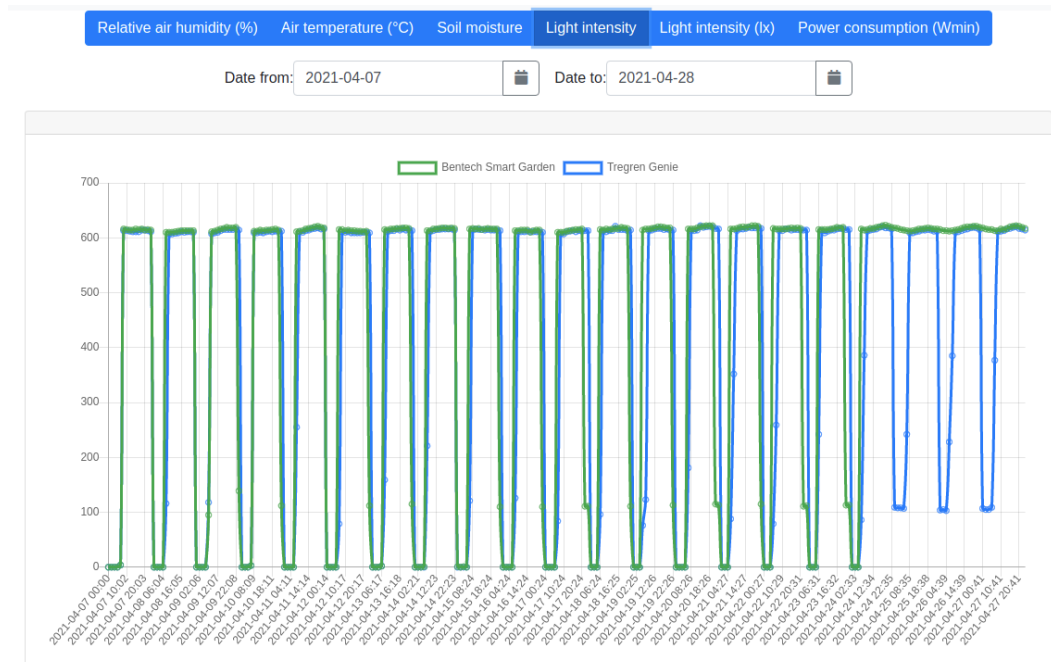
Prozatím byly hodnoceny ukazatele, které měly oba květináče společné, neboť se měřící zařízení i oba květináče nacházely na stejném místě. Následující graf nám ukazuje první charakteristiku, kde se hodnoty rozcházejí - vlhkost půdy. Podle specifikací senzoru, které je možno nalézt na stránkách s dokumentací k sensorům Seeed, se měla hodnota pohybovat v rozmezí 0 (pro absolutní sucho) až po cca 900 (voda). Během testování a následného měření se ovšem ukázalo, že senzor vrací hodnoty až do výše 2000 a rozhodně se nedrží rozmezí uváděných v dokumentaci. Bylo proto nutné lehce přehodnotit i naměřená data.

U modelu Bentech je možno vidět pomalý a stabilní pokles vlhkosti, který je zapříčiněn postupným ubýváním vody z nádržky květináče. Květináč nedisponuje žádným systémem postupného nebo periodického zavlažování a je tak spíše vhodnější pro pěstování hydroponickým způsobem.

Součástí modelu od firmy Tregren je vodní pumpa, která má v pravidelných intervalech vhánět vodu z nádržky na horní platformu s pěstovanou rostlinou. Bohužel skrz celý trvání experimentu pumpa nefungovala. I po pokusu o opravu, který se zdánlivě vydařil, nepumpovala vodu a proto bylo nutné bazalku zavlažovat manuálně. Na křivce tudíž můžeme vidět dva jasně rozpoznatelné „zuby“, ve kterých se vlhkost po zavlažení zvedla. Celkově po dobu experimentu se dařilo držet bazalku rozumně zavlaženou a ukázalo se, že periodické zavlažování nakonec nejspíše svědčilo rostlince více.

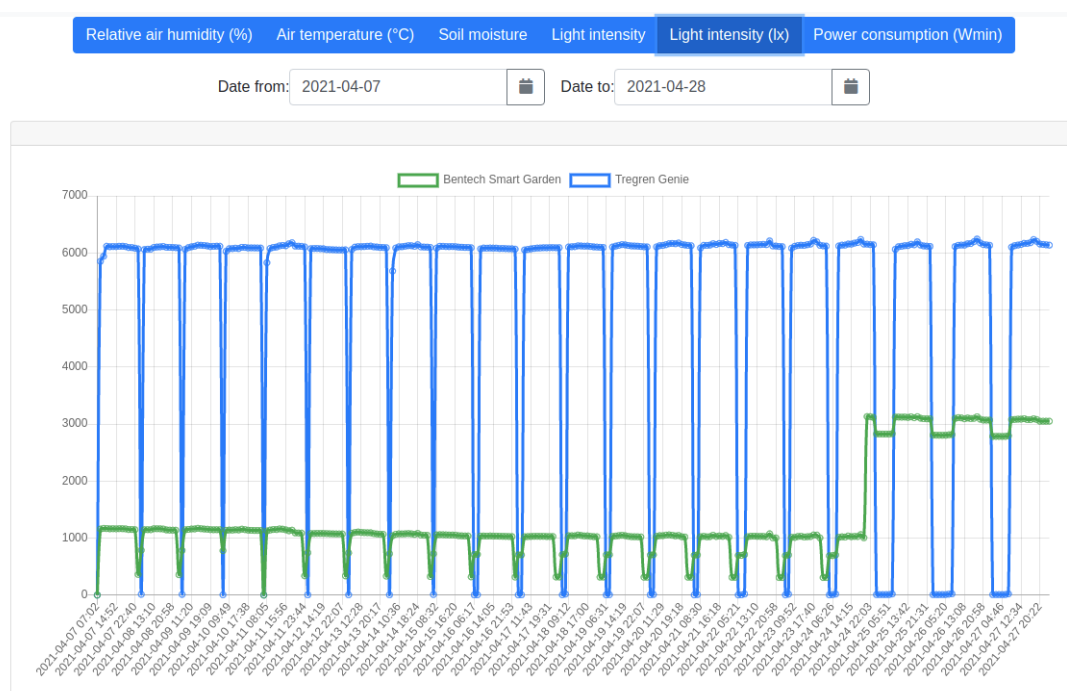
Senzor je tedy možno hodnotit jako funkční, ovšem je nutné se vyrovnat s odchylkou funkčnosti proti specifikaci od výrobce.

## 4.4 Intenzita světla



■ **Obrázek 4.4** Graf intenzity světla obou testovaných květináčů za měřené období (7. 4. 2021 – 28. 4. 2021)

Intenzita světla je vyjádřena dvěma grafy, byla totiž měřena dvěma rozdílnými senzory. Fotorezistor měřil intenzitu světla jako číselnou hodnotu s maximem lehce pod cifrou 620. Během testování se ovšem ukázalo, že nejvyšší hodnotu dosahuje i při relativně mírném světle, tudíž by



■ **Obrázek 4.5** Graf intenzity světla obou testovaných květináčů v luxech za měřené období (7. 4. 2021 – 28. 4. 2021)

nemusel být graf, porovnávající dva různé zdroje světla, dostatečně vypovídající.

Proto byl zakoupen senzor druhý, luxmetr, který převádí intenzitu na jednotku lux.

$$lx = cd * sr \div m^2$$

Jeho relativně široký rozsah měřených hodnot 0 - 65535 lux poskytl dostatečné rozlišení pro oba světelné zdroje a pro jejich různé vlnové délky světla.

U těchto grafů je možné poprvé si všimnout nedokonalosti květináče od firmy Bentech. V manuálu k tomuto modelu je specifikováno, že střídání světla a tmy je v cyklu 16 hodin světlo, 8 hodin tma. Druhý model takový zabudovaný systém nemá, proto bylo zvoleno jako řešení periodické zapínání a vypínání chytré zásuvky, které tuto funkcionalitu simulovalo. Zapínání, respektive vypínání bylo načasováno každý den na 8:00, resp. 24:00 a spouštěno pomocí cronu přímo z řídicí jednotky měřící soustavy. V druhém květináči bylo světlo zapnuto manuálně, přesně v 8:00, čímž započal onen střídavý cyklus řízený elektronikou květináče. Na začátku měřeného období je proto vidět, že křivky se téměř překrývají a květináče jsou synchronizované. Postupem dní se ovšem křivky začínají povážlivě rozjíždět a je vidět, že model Bentech začal časy zapínání a vypínání předbíhat (křivka se proti Tregrenu posouvá doleva). V posledních cca třech dnech dokonce model Bentech vůbec nevypínal, což se stalo po přepnutí druhu světla (květináč má na výběr tři různé druhy), který byl iniciován z důvodu neideálního vývoje rostlinky a pokusu zkusit stav napravit rozdílnou vlnovou délkou světla.

Této změny si je možno detailněji všimnout na grafu, kde se intenzita světla uvádí v jednotce lux. Bílé světlo, emitované led jednotkou u modelu Tregren se pohybuje kolem hodnoty 6000 luxů, červeno modré světlo modelu Bentech dosahovalo hodnoty kolem 1100 luxů, po přepnutí na taktéž bílé světlo se intenzita u Bentechu zvedla k hodnotě lehce nad 3000 luxů. Vidíme tedy, že dioda u květináče Tregren emitovala intenzivnější světlo a i to mohlo později přispět k

výsledku, kdy si rostlinka v květináči od firmy Tregren vedla lépe. Pro měření intenzity různých světelných zdrojů je pak lepší sáhnout po luxmetru, nebo alespoň najít fotorezistor s větším rozsahem, neboť pro tuto práci zvolený kus v podstatě jen naměřil extrémy (tma a maximální světlo).

## 4.5 Spotřeba elektrické energie



■ **Obrázek 4.6** Graf spotřeby elektrické energie obou testovaných květináčů za měřené období (7. 4. 2021 – 28. 4. 2021)

Poslední naměřenou provozní charakteristikou je spotřeba elektrické energie. Použité chytré zásuvky, které toto měření podporují, vrací hodnotu v jednotce Wmin.

$$Wmin = W * min = W * s \div 60 = W * h * 60$$

U modelu Bentech, který po téměř celou dobu měření svítil v červeno-modrém módu se spotřeba pohybovala kolem 2,9 Wmin. Na konci měřeného období, po přepnutí na bílé světlo se spotřeba vyhoupla na 5,6 Wmin. Ve srovnání s květináčem Tregren, který se stabilně pohyboval se svým bílým světlem okolo hodnoty 6,2 Wmin, tedy vychází poměr lx / Wmin ve prospěch Tregrenu. I u bílého světla model od Bentechu dosahoval pouze poloviční intenzity, zatímco ve spotřebě se na Tregren téměř dotáhl. Důležité je také zmínit, že spotřeba květináče od Bentechu nebyla nikdy nulová, neboť stand-by mód konzumoval necelých 0,5 Wmin.

## Kapitola 5

# Testování

Testování je důležitou součástí vývoje softwaru a testován byl i použitý hardware. Testování je rozděleno podle toho, jaké části implementace se týká. Součástí práce je tedy:

- Automatické testy Frontendu aplikace - testy v Angularu pomocí zabudovaného testovacího prostředí Karma
- Automatické testy Backendu aplikace - různé druhy testů v php frameworku Symfony za pomoci knihovny pro testování php aplikací phpunit, statická analýza a coding standards
- Manuální testování software pro řídicí jednotku
- Manuální testování hardwaru

### 5.1 Testování frontendu

Frontend webové aplikace, psaný v Angularu, obsahuje pár jednoduchých testů, využívajících integrovaného testovacího prostředí Karma, což je velmi užitečné, neboť narozdíl od phpunit u backendu nemusí developer nic doinstalovat manuálně, vše má naservírováno již se skeletem aplikace, kterou vytvoří Angular CLI příkazem

```
$ ng new <nazev-nove-aplikace>
```

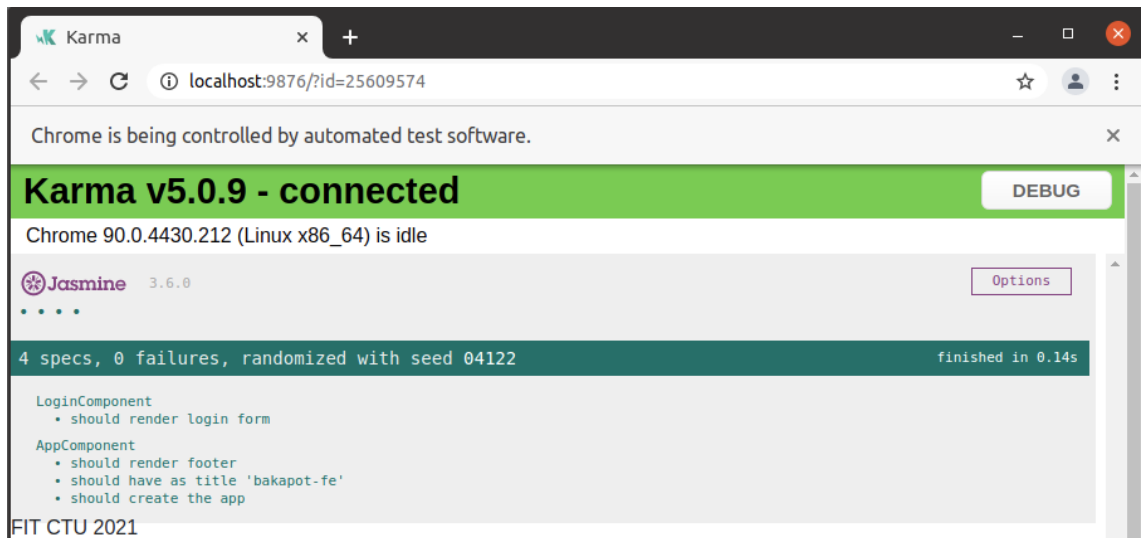
Při vývoji samotném pak platí, že pokud developer nechává nové komponenty generovat pomocí

```
$ ng new component <nazev-komponenty>
```

Angular mu automaticky vygeneruje i soubor se základem pro test této nové komponenty. Proto je možno říct, že samotný framework jde testování velmi proaktivně naproti, neměl by tedy být problém i používat Angular pro tzv. test-driven development, kde je prvním krokem procesu vývoje definice funkcionality a následně napsání testu, který tuto funkcionality ověřuje. Samotné testy se pak spouští pomocí příkazu

```
$ ng test
```

který zkompiluje kód, spustí ovládanou instanci prohlížeče a vypíše výsledky testů. V případě změny kódu za běhu testu funguje automatické překompilování, jako v případě vývoje Angular aplikace. Samotné testy jednotlivých komponent se nacházejí v souborech s koncovkou .spec.ts.



■ **Obrázek 5.1** Okno se spuštěným testem pro frontend aplikace, výsledky a čas testování. Je možno si všimnout i hlášky, že Chrome je kontrolován automatizovaným testovacím softwarem.

## 5.2 Testování backendu

„Neviditelná spojka“ mezi měřicím zařízením a frontendem aplikace, backend napsaný v php za pomoci frameworku Symfony, je testována díky knihovně phpunit. Jedná se o velmi populární testovací knihovnu, použitelnou pro jakoukoliv php aplikaci. Narozdíl od Angularu musí proběhnout manuální instalace knihovny, ovšem díky balíčkovému správci Composer je to záležitost jednoho příkazu. Samotné testy na backendu aplikace jsou umístěny ve složce tests a jsou tedy narozdíl od frontendu jasně separované od zdrojového kódu. Dále jsou rozdělené do složek podle úrovně testů:

- Unit - jednotkové testy, které se zaměřují na vyzkoušení functionality nejmenších jednotek kódu, jako jsou funkce nebo metody tříd.
- Integration - integrační testy se zabývají složitějšími interakcemi kódu, je typicky zapojeno více tříd a je možné volat i služby ze service kontejneru Symfony
- Application - obsahuje komplexní testy, založené na posílání síťových požadavků a kontrole, zda odpověď odpovídá očekávanému výstupu

Protože jsou při testování ukládána do databáze data, je potřeba spustit větší sekvenci příkazů - pro vytvoření testovací databáze, pro vytvoření jejího schématu a konečně pro vložení testovacích datových sad. Proto byl pro potřeby práce vytvořen bash skript, který se o tyto úkony pokaždé stará a před každým spuštěním testů kontroluje zda je vytvořena databáze a maže z ní data. Jediné, co je nutné před samotným spuštěním testů provést je nastavení proměnné prostředí, díky které se aplikace do databáze připojí.

```

> purging database
> loading App\DataFixtures\UserFixtures
PHPUnit 9.5.6 by Sebastian Bergmann and contributors.

Testing
.....
Time: 00:00.425, Memory: 22.00 MB

```

■ **Obrázek 5.2** Výstup testovací knihovny phpunit, bez chyb

Nedílnou součástí dobře testovaného a bezpečného kódu jsou také utility pro jeho statickou analýzu a čistotu kódu. Statická analýza souhrnně označuje techniky pro automatické hledání chyb ve zdrojovém kódu, aniž by tento kód byl spuštěn. Jako čistý pak můžeme označit takový kód, který je ideálně takzvaně self-documented, dobře čitelný a obecně dodržuje jisté standardy (coding standards). Pro statickou analýzu a udržení čistoty kódu backend aplikace jsou zavedeny dvě knihovny:

- **PHPStan** - Jedná se o knihovnu pro testování kódu bez jeho spuštění, kontroluje například typování parametrů, proměnných, návratových hodnot funkcí, detekuje nedefinované proměnné a případy, kdy by proměnná, kvůli podmíněnému rozvětvení programu, nemusela být definována, umí najít nedosažitelné části kódu nebo zbytečné porovnání (always true condition) a spoustu jiných chyb. Dá se konfigurovat přísnost, se kterou jsou chyby odhalovány a pro starý kód se dá nakreslit „základní čára“ (baseline), kdy se chyby starého kódu v analýzách ignorují, pokud by jich bylo tolik, že by se jejich náprava nevyplatila zpětně dělat.
- **PHP\_CodeSniffer** - Knihovna se řídí uživatelem dodanými pravidly, tzv. coding standards, které určují, jaké prohřešky proti čistotě kódu jsou a nejsou akceptovatelné. Jedná se o různá odsazení, mezery, maximální délku řádků, ale i odstranění nepoužívaných importů nebo dodržování camelCase / PascalCase a jiných pojmenovávacích konvencí. Druhá část této knihovny, phpcbf (Code Beautifier and Fixer), se potom stará o automatickou opravu všeho, co opravit bez zásahu uživatele jde. Jedná se hlavně o odsazení, mezery, chybějící declare statementy a doplnění typehintů, kde je zcela zřejmé o jaký typ se jedná.

```

mikoun@mikulas-ntb:~/FIT/bakapot-docker/symfony$ vendor/bin/phpstan analyse
Note: Using configuration file /home/mikoun/FIT/bakapot-docker/symfony/phpstan.neon.dist.
37/37 [██████████] 100%

-----
Line   src/DataFixtures/UserFixtures.php
-----
21     Method App\DataFixtures\UserFixtures::load() has no return typehint specified.
-----

```

■ **Obrázek 5.3** Výstup knihovny PHPStan nad zdrojovým kódem nám říká, že je nutno doplnit nativní typ vrácené hodnotě, dále ukazuje v jakém souboru a na jakém řádku.

```

mikoun@mikulas-ntb:~/FIT/bakapot-docker/symfony$ vendor/bin/phpcs
....E....EE. 12 / 12 (100%)

FILE: src/DataFixtures/UserFixtures.php
-----
FOUND 1 ERROR AFFECTING 1 LINE
-----
 21 | ERROR | [x] Method \App\DataFixtures\UserFixtures::load() does not have void return type hint.
    |       |      (SlevomatCodingStandard.TypeHints.ReturnTypeHint.MissingNativeTypeHint)
-----
PHPCBF CAN FIX THE 1 MARKED SNIFF VIOLATIONS AUTOMATICALLY

```

■ **Obrázek 5.4** Výstup knihovny PHP\_CodeSniffer nad zdrojovým kódem nám tvrdí to samé, co PHPStan. Je proto vidět, že knihovny se občas i doplňují nebo dokonce překrývají. CodeSniffer nám navíc nabízí, že tento neduh na kódu umí automaticky opravit (poslední věta výstupu)

### 5.3 Testování senzorů a měřícího zařízení

Poslední část testování byla zaměřena na software nahraný na řídicí jednotce měřící soustavy a samotné senzory, které jsou k řídicí jednotce připojené. Zde se testování překlátilo do více manuální práce, automatické testy zde svým způsobem nedávají moc smysl.

Začátek byl jasný, bylo nutné vyzkoušet funkčnost všech senzorů, zda a jaké vrací hodnoty při různých podmínkách. Jako první se na řadu dostal senzor vlhkosti půdy. Propojení s hatem na řídicí jednotce proběhlo v pořádku a senzor po stažení python knihoven od firmy Seeed opravdu posílal hodnoty. Bylo ovšem zjištěno, že při zasunutí do hlíny, koupené speciálně jako substrát pro bylinky, senzor vrací hodnoty vyšší než ve vodě, a to dokonce téměř dovnásobně. Hodnoty se měly pohybovat v rozmezí 0 až cca 900, avšak v hlíně senzor vracel hodnoty kolem 1600 a výš. S povrchem jako byla suchá lidská kůže nebo voda ve skleničce, se hodnoty do rozmezí vešly. Proto bylo rozhodnuto, že senzor vadný není, jen má zřejmě půda velmi dobré vodivé vlastnosti.

Další byl senzor intenzity světla, ten pracoval správně. Ve tmě posílal hodnotu blízkou nule, na světle stoupala hodnota nad 600. Tato maximální hodnota byla naměřena ovšem už na mírném světle a senzor tedy nebyl schopen dostatečně rozlišit intenzitu, neboť jeho fotorezistor byl příliš citlivý. Proto byl zakoupen luxmetr, který sliboval větší rozsah měřených hodnot. Ten byl vyzkoušen pod přímým bílým světlem z obou květináčů a i tak měřil hodnotu jen zhruba v desetině jeho škály. Proto byl určen jako primární senzor pro měření intenzity světla. Poté přišla řada na senzor teploty a vlhkosti vzduchu. Zde nebylo moc nad čím přemýšlet a jakmile senzor začal vracet hodnoty, byly porovnány s teploměrem a barometrem pověšeným na zdi místnosti. U relativní vlhkosti se hodnoty lišily maximálně o nízké jednotky a o cca jeden stupeň Celsia u teploty, proto i u tohoto senzoru bylo rozhodnuto o jeho funkčnosti a vhodnosti.

Posledním testovaným kouskem hardware byla chytrá zásuvka, zde byl postup o něco složitější. Nejdříve se musela zásuvka zapojit a začít vysílat WiFi signál, aby se k ní dalo připojit a nakonfigurovat jí jako WiFi klienta, nikoliv jako přístupový bod. Zde ovšem nastal první problém. Aby se mohla zásuvka připojit k Raspberry Pi Zero, muselo Pi vysílat WiFi jako přístupový bod. Sama řídicí jednotka ovšem musela být připojena k internetu, aby mohla komunikovat s backendem aplikace. Tento problém se nakonec po delším hledání podařilo vyřešit za pomoci článku, který bod po bodu vysvětloval postup takové konfigurace. [20] Jakmile bylo toto dvousměrné připojení funkční a bylo možné, aby se zásuvka k Raspberry připojila, byly zkušebně posílány dotazovací i ovládací (vypnutí / zapnutí) požadavky.



Jako poslední část bylo nutné testovat, zda se spouští cronem buzené skripty pro logování dat na vzdálený server a počkat poté na první půlnoc, kdy měla opět díky cronu jedna z chytrých zásuvek vypnout přívod energie (ke květináči Tregren, který nemá řídicí elektroniku pro periodické svícení). To zprvu vedlo k odhalení, že je v Raspberry nastavená jiná časová zóna, ale druhý den už světlo vypnulo přesně o půlnoci.

Testování tudíž bylo úspěšně dokončeno a celý systém od té chvíle byl považován za funkční celek.



■ **Obrázek 5.5** Oba produkční vzorky a mezi nimi právě testované měřicí zařízení. Zde je ještě vidět že nejsou zapojeny všechny senzory, stav je ještě před zasetím bazalky



## Kapitola 6

# Závěr

Cílem práce po analytické stránce bylo prozkoumat trendy v oblasti chytrých květináčů a jimi podporovaných bylin. Začal jsem tedy rešerší zabývající se rostlinami, které chytré květináče znají a umí si s nimi poradit. Zjistil jsem, že velmi často se jedná o různé bylinky nebo zeleninu používané při vaření, ovšem i květiny zde mají zástupce. Mezi veškerými rostlinami ale drtivě zvítězila jedna, která se často objevovala na produktových fotografiích nebo byla přednostně zmiňována v informacích o modelu. Touto královnou segmentu chytrých květináčů je bazalka, proto jsem jí vybral jako květinu vhodnou pro experimentální pěstování.

Následoval výběr produkčních vzorků květináčů, na kterých se mělo pěstování realizovat. Zadání sice mluvilo v jednotném čísle, ale já se rozhodl proti sobě postavit květináče dva, abych zjistil jak moc se můžou modely od dvou různých firem na jednom a tom samém úkolu lišit. Porovnal jsem proto různá existující a prodávaná řešení a nakonec vybral modely od firem Bentech a Tregren. Cenově šlo spíše o nižší kategorii, ovšem i tak oba výrobci deklarovali, že jejich zástupci jsou květináče chytré a pěstování maximálně usnadní a pomůžou s ním. Co se týče obou modelů, ani jeden nedisponuje pokročilými chytrými funkcemi, jako je hlídání pH půdy, monitorování vlhkosti půdy nebo upozornění na nedostatek vody v nádržce. Zcela pak chybí jakákoliv softwarová podpora, jako například mobilní nebo desktopová aplikace pro sledování údajů o rostlině, ukazatelů prostředí, popřípadě s upozorněními na nedostatek vody či nedostatečné hnojení půdy.

Po výběru produkčních vzorků jsem sestavil měřicí aparaturu, otestoval senzory, naprogramoval webovou aplikaci, zasadil bazalku a pustil se do experimentu s pěstováním. Během toho jsem odhalil několik nedostatků u obou použitých květináčů.

Květináč od firmy Bentech vykazoval odchylky od v příručce jasně stanoveného cyklu 16 hodin rozsvíceno / 8 hodin zhasnuto. Je také škoda, že se interval nijak nedal přenastavit, v případě specifické potřeby pro jiné typy rostlin. Co je naopak dobré přičíst Bentech Smart Garden k dobru, je výběr ze tří různých intenzit a barevných provedení osvětlení. Samotná konstrukce je spíše vhodnější pro hydroponické pěstování, neboť není přítomna vodní pumpa, rozprašovač, nebo nějaký jiný systém pro automaticky dávkovanou závlahu.

Model od firmy Tregren také nebyl ve všem stoprocentní. Osvětlení květináče je řešeno pouze jedním druhem bílého světla. Samotné osvětlení je řešeno jen jako „hloupá“ lampička, není zde vůbec přítomen interval střídání svícení a tmy. Tudíž jako jediné takové řešení je pak nutno zvolit nějakou formu chytrého řešení zásuvky, do které je osvětlení Tregren květináče zapojeno. Co se týče závlahy, je zde přítomna vodní pumpa, jejíž napájení je ovšem řešeno pomocí čtyř tužkových baterií, což je dle mého soudu velmi nekonzistentní a nesytematické. Proč by měl z květináče jít drát do zásuvky kvůli světlu, ale pumpa si měla brát energii z relativně těžko přístupných baterií na spodku květináče? Samotné zavlažování je řešeno peroidickým pumpováním vody z nádržky na plošinu s rostlinou. Bohužel, v zakoupeném Tregren vzorku vodní pumpa fungovala

jen sporadicky. Po relativně dlouhé diagnostice jsem zjistil, že šlo o vadný drátek, propojující řídicí relé a samotnou vodní pumpu. Avšak to samozřejmě nelze považovat za chybu modelu, vada se totiž nejspíše nevyskytuje na všech vyrobených kusech.

Všechny zjištěné nedostatky automaticky vedly k myšlenkám, co vše se dá vylepšit a udělat jinak, efektivněji, důmyslněji, proto vlastně tato zjištění kvituji. Vytvořené měřicí zařízení některé tyto chybějící funkce může zastoupit. Nabízí se buďto cesta vytvoření prototypu fakultního květináče s využitím jeho komponent, nebo by mohl, ve spojení s monitorovací aplikací, která je díky použití dockeru a rozdělení na backend a javascript frontend vysoce přenosná, vzniknout podpůrný systém s prakticky neomezenou škálovatelností, co se pěstební plochy / počtu obsluhovaných květináčů týče. Aplikace je od začátku psaná s možností existence více pěstebních jednotek (v případě této práce se jedná o dva květináče), tudíž je možno do ní logovat data z teoreticky neomezeného počtu zdrojů. Je také možné aplikaci do budoucna rozšířit o komponenty aktivní, které by mohly kromě samotného monitorování i spouštět akce, například zavlažování, zapínání a vypínání světel a jiné. Důležitou vlastností vytvořené aplikace je také to, že se jedná o aplikaci webovou, tudíž jednoduše zobrazitelnou na prakticky jakémkoliv zařízení s prohlížečem a přístupem na internet. Odpadá tedy nutnost instalace nativní aplikace, stejně jako i případné zvýšení nákladů na údržbu těchto separátních nativních řešení.

Podařilo se také sesbírat provozní data, která by se v případě konstrukce prototypu mohla využít pro porovnání s daty při provozu navrhovaného modelu. Tím se podařilo zdárně splnit všechny body ze zadání, což považuji za úspěch.

# Bibliografie

1. *TREGREN T3 Kitchen Garden, bílý* — Alza.cz [online] [cit. 2021-02-27]. Dostupné z: <https://www.alza.cz/tregren-t3-kitchen-garden-bily-d5695950.htm?o=3>.
2. *Tregren SmartGardener – Aplikace na Google Play* [online] [cit. 2021-02-27]. Dostupné z: <https://play.google.com/store/apps/details?id=com.monnisoftware.tregren&hl=cs&gl=US>.
3. HAUSCHILD, Stephanie. *Pěstujeme bylinky v bytě: [čerstvé i v zimě]*. Praha: Grada, 2011. ISBN 978-80-247-3814-7.
4. HORYNOVÁ, Anna. *Praktické zahradnictví: ovocnictví - zelinářství*. Praha: Státní zemědělské nakladatelství, 1966.
5. BIELIKOVÁ, Stella. *Hydroponický kvetináč*. Praha, 2019. Bakalářská práce. ČVUT v Praze, Fakulta architektury, Ústav průmyslového designu.
6. FAKTOROVÁ, Michaela. *Nekonve*. Praha, 2020. Magisterská práce. ČVUT v Praze, Fakulta architektury, Design. something.
7. ČEBIŠ, Vladimír. *Automatický systém pro pěstování rostlin*. Praha, 2018. Bakalářská práce. ČVUT v Praze, Fakulta elektrotechnická, katedra mikroelektroniky.
8. AL, Shengyue Fan et. Flower “Dock” - Single Chip Microcomputer-Based Smart Flowerpot. *Information* [online]. 2020 [cit. 2021-03-27]. Dostupné z DOI: 10.1088/1757-899X/750/1/012120.
9. MONK, Simon. *Programming the Raspberry Pi: getting started with Python*. New York: McGraw-Hill, 2013. ISBN 978-0-07-180783-80.
10. UPTON, Eben; HALFACREE, Gareth. *Raspberry Pi: uživatelská příručka*. 2. vyd. Brno: Computer press, 2016. ISBN 978-80-251-4819-8.
11. GRIEF, Sasha; BENITTE, Raphaël (ed.). *State of JS 2020: Front-end Frameworks* [online]. 2020. vyd. 2021-01-13 [cit. 2021-06-13]. Dostupné z: <https://2020.stateofjs.com/en-US/technologies/front-end-frameworks/>.
12. Typescript: Documentation: TypeScript for the New Programmer [online]. [N.d.] [cit. 2021-06-21]. Dostupné z: <https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>.
13. *Databases Market Share Report — Competitor Analysis — Microsoft SQL Server, MySQL, Microsoft Access* [online]. Datanyze [cit. 2021-02-20]. Dostupné z: <https://www.datanyze.com/market-share/databases--272>.
14. *Usage Statistics and Market Share of Web Servers, June 2021* [online]. Q-Success [cit. 2021-06-10]. Dostupné z: [https://w3techs.com/technologies/overview/web\\_server](https://w3techs.com/technologies/overview/web_server).

15. *Web Server Survey — Netcraft News* [online]. Netcraft Ltd, 2021-05 [cit. 2021-06-10]. Dostupné z: <https://news.netcraft.com/archives/category/web-server-survey/>.
16. *Stack Overflow Developer Survey 2020* [online]. Stack Overflow [cit. 2021-06-20]. Dostupné z: <https://insights.stackoverflow.com/survey/2020#most-popular-technologies>.
17. *Symfony releases, notifications and release checker* [online]. Symfony SAS [cit. 2021-06-21]. Dostupné z: <https://symfony.com/releases>.
18. *Účinky vlhkosti na sbírkové materiály* [online]. Slezská univerzita v Opavě, 2015 [cit. 2021-06-21]. Dostupné z: <https://www.slu.cz/slu/cz/projekty/webs/zkvalitneni/vystupy-1/uhv-m0096-preventivni-konzervace/vlhkost>.
19. *Meteostanice - Praha - Kbely, aktuální teplota a počasí, archiv, rekordy — In-pocasi* [online]. InMeteo s.r.o. [cit. 2021-06-15]. Dostupné z: [https://www.in-pocasi.cz/archiv/praha\\_kbely/?detailed\\_typ=srazky&detailed\\_usek=36#daily\\_graph](https://www.in-pocasi.cz/archiv/praha_kbely/?detailed_typ=srazky&detailed_usek=36#daily_graph).
20. CHAHARBAKHSI, Albert. *Raspberry Pi Zero W Simultaneous AP and Managed Mode Wifi* [online]. 2017-09-26 [cit. 2021-02-20]. Dostupné z: <https://blog.thewalr.us/2017/09/26/raspberry-pi-zero-w-simultaneous-ap-and-managed-mode-wifi/>.

# Obsah přiloženého média

src	
├─ bakapot-angular .....	zdrojové kódy frontendu webové aplikace
├─ bakapot-docker .....	zdrojové kódy backendu webové aplikace
├─ bakapot-raspberry .....	zdrojové kódy skriptů, ovládajících měřicí zařízení
├─ thesis .....	zdrojová forma práce ve formátu L <sup>A</sup> T <sub>E</sub> X
text .....	text práce
├─ thesis.pdf .....	text práce ve formátu PDF