

# SEMINÁRNÍ PRÁCE

Předmět: Programování a výpočetní technika

## Monitoring terária

## Terrarium monitoring

Škola: Gymnázium Teplice



Vypracoval: Prokop Parůžek

Konzultant: Ing. Věra Minaříková

Rok: 2021



## **Prohlášení**

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

Prohlašuji, že tištěná verze a elektronická verze práce jsou shodné.

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

V Teplicích dne .....

.....  
Prokop Parůžek

## Poděkování

Děkuji ... za ... (upravte makro \podekovani{}).

Prokop Parůžek

*Název práce:*

## **Monitoring terária**

*Anotace:* Cílem práce je vytvořit automatický systém na sledování teploty a vlhkosti a dalších údajů v teráriu s masožravými rostlinami. Zpřístupnit naměřené údaje online, v podobě grafů, aby uživatel mohl v reálném čase sledovat jak se jeho kytičkám daří. Zároveň je kladen důraz na snadnou rozšiřitelnost o další naměřené hodnoty, či o úplně nové senzory, místnosti.

*Klíčová slova:* Klíčová slova

*Title:*

## **Terrarium monitoring**

*Annotation:*

*Key words:* Key words



# Obsah

<b>Úvod</b>	<b>9</b>
<b>1 Požadavky na řešení</b>	<b>11</b>
<b>2 Analýza problému</b>	<b>13</b>
<b>3 Hardware</b>	<b>15</b>
<b>4 Software</b>	<b>17</b>
4.1 Měřicí stanice . . . . .	17
4.2 Domácí gateway . . . . .	17
4.3 Cloud . . . . .	17
4.4 Zobrazení grafů . . . . .	17
<b>5 Výsledek</b>	<b>19</b>
<b>Závěr</b>	<b>21</b>
<b>Literatura</b>	<b>23</b>
<b>Přílohy</b>	<b>25</b>
<b>A Zdrojový kód</b>	<b>27</b>





# Úvod

Už potřetí zahajuji svůj pokus pěstovat masožravé rostliny, který zatím vždy skončil jejich úhynem. Z toho důvodu jsem se rozhodl začít sledovat prostředí v teráriu, kde je pěstuji, abych mohl v případě úhynu určit z jakého důvodu uhynuly. Přehřáli se, umrzli, uschly... Většinou z důvodu mé nepřítomnosti, kdy jsem je nemohl kontrolovat. Avšak mnohem radši bych byl, kdyby se mi pomocí naměřených údajů podařilo udržet prostředí ve kterém prosperují a v případě náhlé změny mohl zasáhnout v krajním případě i vzdáleně.

Cílem mé práce je navržení systému pro měření v podstatě libovolných hodnot, jejich agregování na jednom místě, s možností zobrazení aktuálních dat, či jejich průběhu v minulosti, či navázáním různých alarmů na kritické hodnoty. Hodnoty by uživatel kontroloval s využitím webové aplikace, které zároveň zajistí snadnou použitelnost na mnoha platformách a přístupnost takřka na celém světě, tedy tam kde je internet.

Výsledkem práce bude samotná realizace řešení, od výběru hardwaru a dalších věcí jako je databáze. . . po samotné sestavení měřicího zařízení, jeho naprogramování a naprogramování aplikace na zobrazení naměřených dat. Výsledný produkt by měl být snadno použitelný a rozšiřitelný o další funkce, možný budoucí vývoj je až aplikace na řízení tzv. chytrého domu. Z tohoto důvodu bude kladen důraz i na zabezpečení, pro zamezení neoprávněného přístupu. Z důvodů urychlení a zlevnění vývoje, nebudu vždy používat nejvhodnější, ale nejdostupnější řešení tj. to které už znám, či u hardwaru to co mám doma.



# Kapitola 1

## Požadavky na řešení

Na celé řešení mám několik požadavků, které postupně detailně popíši tak, aby bylo vše jasné. Poněvadž není nic horšího než nepřesné, či nejasné zadání, protože se pak výsledek špatně hodnotí a celkově upřesňování zadání v průběhu řešení je cesta do pekel. Tyto požadavky jsou nutné, pokud vytvořené řešení bude mít funkce navíc, nevadí to, ale nesmí ovlivňovat tyto zadané parametry.

První požadavek se bude týkat měření. Měřit chci teplotu a vlhkost v teráriu, když bude zvolený hardware umět i něco jiného, klidně to použiji, ale hlavní požadavek je na tyto dvě veličiny. Ohledně frekvence měření chci zachytit denní trendy, ale nepotřebuji data z každé minuty.

Další z požadavků je na ukládání dat. Když už je změřím, tak je chci mít vždy uložená, tedy i při výpadku internetu a podobně. Při výpadku proudu nic nezměřím, takže to není třeba řešit. Co se týče vzdáleného ukládání, nepovažuji za důležité, aby se všechna data propsala do cloudu, tedy i v případě nějakého výpadku se odeslala data co jsem změřil, ale neodeslal. Když přijdu o jedno měření během krátkého výpadku, je mi to jedno a při větším nebo nějaké chybě si toho všimnu a stejně to bude třeba opravit.

Data mám uložena, co s nimi budu dělat? No tak v podstatě bych nemusel dělat nic, pouze si je zobrazit, což je to co požaduji. Avšak hezká by byla možnost zobrazit si nějaké pokročilejší statistiky. Takže volitelně přidávám požadavek na zobrazení různých časových rámců a statistiky k onomu časovému období, například průměrná, nejvyšší, nejnižší či medián teploty a vlhkosti. Případně různé tendence, derivace, co bude v možnostech vybraného nástroje.

Nároky na uživatelské rozhraní v podstatě nemám. Pro správu senzorů je nepotřebné. Stejně moc obměňovat, či přidávat nebudu, a i kdyby Stejně si budu muset napsat obslužný program pro ten či onen. Udělat nějaký obalující systém pro více senzorů, či knihovnu není mým cílem. Pro zobrazení hodnot je důležité, ale ohledně nároků mi bude stačit velmi jednoduché, pokud možno jediná stránka. Avšak případnému rozšíření se nebráním. Další části u nichž by bylo třeba komunikovat s uživatelem mě nenapadají. Možná správa uživatelů s přístupem, ale vzhledem k tomu, že to dělám pro sebe bych to vynechal.

K bezpečnosti bych rád zmínil toto. Bylo by dobré mít komunikaci po celé komunikační trase šifrovanou, ale dokud řešení neobsahuje ovládání čehosi, není to úplně nezbytné. Případný útočník by si tak maximálně zjistil vlhkost... v teráriu nebo by se mohl teoreticky vydávat za teploměr a kazit mi data. To by mohl být problém, takže pokud nevyžadují šifrování, ověření toho kdo posílá data požadují určitě. Zabezpečení však považuji za důležité u samotné webové stránky, která bude vystavena veřejně. Ne že by mi vadilo, že někdo sleduje jak se mají mé kyticky, ale mohlo by se z toho dát odvodit, že je nezáleám tj. jsem pryč a v bytě nikdo není. Pokud si někdo dá tu práci, že mi napíchne připojení, nebo přijede a odposlechne data z domácí sítě, tak má asi i možnosti jak si to, že nejsem doma zjistit jinak. Rozhodně to ale bude složitější než otevření webové stránky.

Pak tu mám poslední z požadavků a tím je cena. Nejraději bych ho neřešil, ale žijeme ve světě kde je tento požadavek velmi důležitý. Takže asi jediné cenové omezení je, abych si to mohl dovolit. Tady bych zopakoval, že cenu budu částečně upřednostňovat před ostatními požadavky. Zejména v případech, kdy mám nějaký hardware doma, i když ne úplně vhodný, ale dostačující. Tak ho upřednostním před vhodnějším, který nemám a musel bych ho draze shánět. Otázku ceny budu podrobněji řešit u výběru hardwaru.

# Kapitola 2

## Analýza problému

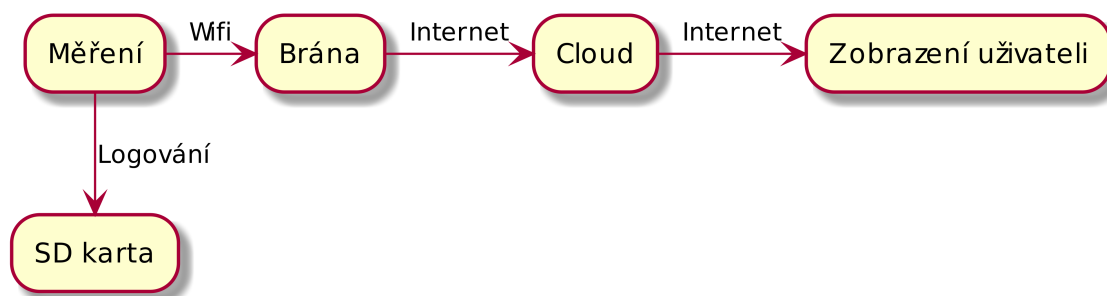
Zde bych rád rozebral jak budu jednotlivé požadavky řešit. Nastíním jak bude celkové řešení vypadat, strukturu... Též bych zde rád představil kde vlastně budu měřit a zpracovávat data.

Ohledně měření bych zatím zmínil jen frekvence o které myslím myslím, že pro mé účely bude stačit měřit jednou za čtvrt hodiny, to je 96 měření za den. Neodchytím tím sice drobné výkyvy v rámci minut, ale cílem je zachytit dlouhodobý průběh hodnot v rámci dne, kdy mne tak drobné výkyvy nezajímají. Pro tento účel by se čtvrt hodiny mohlo zdát možná až jako příliš často, ale já bych to tak nechal z důvodu zjemnění denních grafů, přeci jenom graf se třeba 12 hodnotami nevypadá úplně nejlépe, a také z důvodu odchycení případných chyb měření nebo náhlých změn, například při zalévání atd.

Co se týče uložení dat, abych zajistil stoprocentní jistotu, že se data uloží, budu je ukládat přímo v místě měření, tím myslím, že počítač, který bude mít na starosti měřit, bude zároveň naměřená data hned ukládat na SD kartu, nebo tak někam. Bude to takový log měření, který mi umožní obnovit data nezapsaná do cloudu z důvodu nějaké chyby, případně mi umožní zjišťovat kde vlastně chyba nastala, a může pomoci i s řešením. Dále myslím, že data bude stačit ukládat klidně pouze na jedno místo v cloudu, poněvadž ten sám o sobě pokud je kvalitní je výborně zálohovaný, pokud o data v něm nějakým nedopatřením přijdu, nějak mě to neovlivní, bude to škoda, avšak závažné důsledky to mít nebude a navíc tím, že nebudu řešit zálohy a rozkopírovávání dat, či jejich konzistenci na různých místech se celé řešení výrazně zjednoduší.

Ted' se dostávám k samotnému data flow, tedy jak a kam mi potečou data co naměřím. Začnu u senzoru, ten změří data a pošle je do obslužného počítače, to může být v podstatě cokoli se schopností ovládat senzor, ukládat data a schopností poslat data přes wifi. Ten data zalogue na perzistentní úložiště a pošle je na centrální bránu pomocí wifi sítě, což mi přijde nejjednodušší, nemusím nikde tahat kabely, či řešit jiné bezdrátové technologie, poněvadž wifi síť má v místě měření dostatečné pokrytí. Ted' se dostávám k takovému kontroverznímu prvku celého flow, a tím je centrální brána. To je v podstatě počítač, který jediný co dělá je přeposílání dat ze senzorů někam jinam, dal by se tedy úplně odstranit s tím že měřicí stanice by data odesílala

přímo do internetu, ale já ji zahrnul z těchto důvodů. Díky tomu, že s internetem komunikuje pouze jedna stanice nemusím na ostatních řešit jejich autentizaci vůči cloudu, či různé SSL certifikáty a podobně. To mi umožní na jejich místech mohu nasadit mnohem jednodušší zařízení. Dále mi to umožní přístup k datům doma i bez internetu, kdybych si je chtěl nějak zobrazit... A Také to zjednoduší ladění celého systému, kdy například programy mohu testovat u sebe na počítači kdy data budu brát z centrální brány a nebudu muset vůbec zasahovat do kódu v senzoru. No a to je celé z brány pošlu data do cloudu a tam jejich cesta končí, pokud tedy vynechám cestu z cloudu za účelem jejich zobrazení.



Obrázek 2.1: Takto přibližně potečou data

Uživatelské rozhraní pro zobrazení hodnot jsem se rozhodl z důvodu co největší přenositelnosti ji implementovat jako webovou stránku. Takže celá jeho logika a vykreslování bude řešená v javascriptu a až na klientském zařízení, cloud použiji pouze na to, abych z něj vytáhl potřebná data a samozřejmě na hostování celé aplikace. Na této stránce určitě zobrazím graf vývoje změřených hodnot, myslím že v základu by mohlo stačit tak posledních 48 hodin, ale asi bych přidal možnost i delších časových úseků. Z dalších údajů bych si zobrazil tak možná medián naměřených hodnot a možná průměr, ale další údaje mi už přijdou zbytečné. Další analýzy dělat nebudu, spíše bude sledovat jak se kytičkám daří a případně je po nasbírání zkušeností doplním.

Co se otázky bezpečnosti týče, tak vynechám šifrování na cestě od senzoru do brány, zejména z důvodu jednoduššího ladění a implementace, avšak co na této cestě doplním je elektronický podpis zprávy, aby se mi nikdo nefušoval do komunikace. Zabezpečení komunikace s cloudem už budu řešit prostředky té dané služby i co se zobrazení hodnot týče.

# Kapitola 3

## Hardware

Konečně se dostáváme k něčemu reálnému. Zde popíši hardwarovou část řešení tj. Co vlastně bude měřit, čím budu měřit...

Možných základů pro měřicí stanici, jednodeskových počítačů, je dnes na trhu mnoho od různých osmibitů, až po počítače na architektuře ARM, které dosahují výkonu srovnatelného s mobilními telefony. Já jsem pro mé řešení zvolil Raspberry Pi ve verzi 2 a to z několika důvodů. Jednak ho mám k dispozici a dále mi nabízí běžící Linux a tudíž za mne řeší spoustu problémů, od síťové komunikace, po třeba synchronizaci času. Navíc mám k dispozici spoustu digitálních pinů pro připojení různých senzorů, též mám vyřešené i místní úložiště, data se mohou ukládat na SD kartu ze které běží celý systém a výrazně to zjednodušuje vývoj, poněvadž mohu nahrávat nové verze programů vzdáleně, a i vzdáleně sledovat jejich běh, což je pro mě výhodné, neb terárium nemám v pokoji, kde programuji. Samozřejmě že toto řešení má i své nevýhody. Například v případě, že bych chtěl měřit analogové hodnoty, bych musel dokupovat převodník z analogového signálu na digitální, či v případě nutnosti rozšíření na více míst, by to nebylo ekonomicky výhodné, přeci jen Raspberry Pi stojí kolem 1000 Kč. Nebo pokud bych chtěl zařízení napájet z akumulátoru, tak též s odběrem kolem půl ampéru to nebude to pravé ořechové. Avšak v případě zmíněných problémů mi zvolené celkové řešení umožňuje poměrně komfortně změnit základ stanice na něco vhodnějšího, za předpokladu, že se zvolená deska zvládne připojit na lokální síť. Například mohu použít oblíbenou desku ESP8266 či ESP32, které se cenově pohybují v řádu stokorun, pinů mají dostatek, disponují Wifi čipem a umožňují použití nízko odběrových módů při běhu na baterii.

A teď součást bez které by zbytek byl k ničemu, samotné senzory. Pro měření v samotném teráriu jsem zvolil BME280. Jedná se o senzor od firmy Bosh pro měření teploty, vlhkosti a atmosférického tlaku, vzhledem k tomu, že se dodává pouze v SMD variantě, jsem zvolil modul, který obsahuje tento senzor, potřebné součástky kolem a má vyvedenou I2C sběrnici, pro snadné připojení k řídicí desce. Dal by se též použít libovolný jiný senzor obdobných parametrů, ale tento má příjemnou cenu a velmi dobrou přesnost měření, a navíc se díky I2C snadno propojí se zbytkem systému.

Za účelem případné další analýzy, jsem umístil pár senzorů i mimo terárium,

abych mohl sledovat závislost teploty vně a uvnitř. Tady už nejde tolik o přesnost, takže jsem použil senzory co jsem našel doma. Na měření teploty jsem použil senzor od firmy DS1820. Jde o jednoduché a levné čidlo, moje je v pouzdru TO-92, to se používá třeba taky pro tranzistory, OneWire sběrnici, speciální sběrnice od firmy Dallas, která potřebuje pouze tři dráty a umožňuje i použití pouze dvou a digitálním měřením teploty s přesností 0.5 °C.

A jako poslední jsem použil oblíbené čidlo teploty a vlhkosti DHT11. Jde o čínský výrobek v modré krabičce se čtyřmi vývody, z nichž jeden je neaktivní. Čidlo používá vlastní sběrnici, které však též stačí tři dráty, jako výše zmíněné OneWire. Senzor teploty jsem zdvojit, poněvadž toto čidlo je velmi levné a bohužel též nepřesné, rozlišení 1 °C při možné odchylce až 2 °C, není úplně ideální, proto jsem ho doplnil výše zmíněným teploměrem. U vlhkosti je to podobné, ale vzhledem k tomu, že se jednak hůře ovlivňuje a také se pohybuje ve větším intervalu než teplota mi odchylka 5 % zas tolik nevadí.



# Kapitola 4

## Software

4.1 Měřicí stanice

4.2 Domáci gateway

4.3 Cloud

4.4 Zobrazení grafů



## Kapitola 5

### Výsledek



## Závěr



# Literatura

- [1] TIŠŇOVSKÝ, P. *Použití message brokeru NATS* [online]. březen 2019 [cit. 15. listopadu 2020]. Dostupné na: <https://www.root.cz/clanky/pouziti-message-brokeru-nats/>.
- [2] TIŠŇOVSKÝ, P. *Komunikace s message brokery z programovacího jazyka Go* [online]. březen 2019 [cit. 15. listopadu 2020]. Dostupné na: <https://www.root.cz/clanky/komunikace-s-message-brokery-z-programovaciho-jazyka-go/>.
- [3] TIŠŇOVSKÝ, P. *NATS Streaming Server* [online]. duben 2019 [cit. 15. listopadu 2020]. Dostupné na: <https://www.root.cz/clanky/nats-streaming-server/>.
- [4] TIŠŇOVSKÝ, P. *Tvorba grafů v jazyce Go: kreslení ve webovém klientu* [online]. leden 2020 [cit. 15. listopadu 2020]. Dostupné na: <https://www.root.cz/clanky/tvorba-grafu-v-jazyce-go-kresleni-ve-webovem-klientu/>.
- [5] LOURME, O. *Post 1 of 3. Our IoT journey through ESP8266, Firebase and Plotly.js* [online]. srpen 2018 [cit. 15. listopadu 2020]. Dostupné na: <https://medium.com/@o.lourme/our-iot-journey-through-esp8266-firebase-angular-and-plotly-js-part-1-a07db495ac5f>.
- [6] LOURME, O. *Post 2 of 3. Our IoT journey through ESP8266, Firebase and Plotly.js* [online]. říjen 2018 [cit. 15. listopadu 2020]. Dostupné na: <https://medium.com/@o.lourme/our-iot-journey-through-esp8266-firebase-angular-and-plotly-js-part-2-14b0609d3f5e>.
- [7] LOURME, O. *Post 3 of 3. Our IoT journey through ESP8266, Firebase and Plotly.js* [online]. prosinec 2018 [cit. 15. listopadu 2020]. Dostupné na: <https://medium.com/@o.lourme/our-iot-journey-through-esp8266-firebase-angular-and-plotly-js-part-3-644048e90ca4>.





## Přílohy



Příloha A

Zdrojový kód