



STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
OBCHODNÍ AKADEMIE  
JAZYKOVÁ ŠKOLA  
FRÝDEK~MÍSTEK



Příspěvková organizace  
Moravskoslezského kraje

# Zavlažovací systém PlantHub

Filip Sikora, Jakub Vantuch

soutěž SOČ 2022

## **Prohlášení**

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ 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) ve znění pozdějších předpisů.

Ve Frýdku-Místku dne 4. 3. 2022

.....

## **Prohlášení**

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ 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) ve znění pozdějších předpisů.

Ve Frýdku-Místku dne 4. 3. 2022

.....

## **Anotace**

PlantHub je automatický zavlažovací systém s WUI. Jádrem našeho systému je mikropočítač RPi s procesorovou architekturou ARM, GPIO a možností připojení pomocí ethernetu. Vybrali jsme si jej, protože kombinuje malou velikost a vyšší výpočetní sílu než Arduino. Musí totiž zvládnout řídit všechny senzory, ukládat data do databáze a zároveň hostuje i samotnou webovou aplikaci. Systém PlantHub dále získává informace o teplotě, vlhkosti a tlaku vzduchu a promítá je ve svém WUI. Ve stejné chvíli naměřená data ukládá do databáze v periodě 4 hodin. Jelikož voda časem z nádrže dojde systém PlantHub snímá stav hladiny vody v nádrži a včas upozorní, že je třeba doplnit vodu.

## **Klíčová slova**

zavlažování; automatizace; statistika; živě; RaspberryPi; uživatelské rozhraní

# Obsah

<b>1 Úvod</b>	<b>4</b>
<b>2 Hardware</b>	<b>5</b>
2.1 DHT11 . . . . .	5
2.2 Senzor vlhkosti půdy . . . . .	5
2.3 MCP3008 . . . . .	5
2.4 HC-SR04 . . . . .	6
2.5 Čerpadlo . . . . .	6
2.6 2N2222 . . . . .	6
<b>3 Obvod</b>	<b>7</b>
<b>4 Hlavní program</b>	<b>8</b>
4.1 Postup práce . . . . .	8
4.2 Fáze programu . . . . .	8
4.2.1 Měření . . . . .	8
4.2.2 Periodické ukládání dat . . . . .	8
4.2.3 Controller . . . . .	8
4.2.4 Inicializace . . . . .	9
4.2.5 Zavlažování . . . . .	9
4.2.6 Kontrola . . . . .	9
4.3 Ukládání dat . . . . .	9
<b>5 WUI</b>	<b>10</b>
<b>6 Databáze</b>	<b>12</b>
<b>7 Webový server</b>	<b>13</b>
<b>8 Na čem ještě pracujeme</b>	<b>14</b>
<b>9 Závěr</b>	<b>15</b>
<b>10 Seznam zkratk a pojmů</b>	<b>16</b>
<b>11 Použité nástroje</b>	<b>17</b>
<b>12 Použité zdroje</b>	<b>18</b>
<b>13 Seznam obrázků</b>	<b>19</b>
<b>14 Seznam příloh</b>	<b>20</b>

# 1 Úvod

Naším cílem je návrh, sestavení a naprogramování automatického zavlažovacího systému s WUI a ukládáním naměřených dat do databáze pro pozdější statistiky. Tato stanice pravidelně snímá data ze senzorů měřících teplotu a vlhkost vzduchu, vlhkost půdy a stav hladiny v nádrži, z té pak stanice přečerpává vodu pomocí čerpadla spouštěného tranzistorem. Naměřená data se posílají živě pomocí REST API prostřednictvím protokolu HTTP do WUI a ve stejné chvíli se v periodě čtyř hodin ukládají do PostgreSQL databáze běžící v docker containeru pro pozdější statistiky změn teploty, vlhkosti vzduchu, vlhkosti půdy a počtů zavlažování v časovém rozsahu zobrazitelné v dashboardu WUI. Pro načtení historických dat z databáze používáme GraphQL API, jelikož nabízí šetrnější přístup k datům, vybráním pouze těch záznamů z tabulky, které opravdu využíváme. V případě nedostupnosti serveru by se ve WUI měla pořád načítat živě naměřená data z REST API. Přístup k naměřeným datům a WUI má pouze uživatel lokální sítě, do které je planthub připojen pomocí ethernet kabelu, z čehož vyplývá, že pro správnou funkci stanice je zapotřebí router s přístupem k internetu a DHCP serverem.

Ve WUI hostovaném na naší stanici, a sestavém pomocí javascriptového frameworku React.js, CSS frameworku Tailwind a programovacího jazyku Typescript, který je supersetem javascriptu, podporujícím volitelné statické typování, má uživatel možnost zobrazení statistik jak živě naměřených dat tak dat historických v nastavitelných grafech dashboardu. WUI také nabízí manuální kontrolu nad čerpadlem s časovačem začátku zavlažování a restartem hlavního programu. Nastavitelné je jak WUI tak i samotné faktory zavlažování jako je limit vlhkosti půdy, doba zavlažování nebo limit hladiny vody.

Jednotlivé součásti a moduly stanice jsme vybrali podle finanční dostupnosti a adekvátních technických požadavků na přesnost měření. Zapojení těchto modulů do obvodu jsme prvně provedli ve webové aplikaci EasyEDA, která nabízí jednoduché prostředky pro návrh technických schémat. Tento obvod jsme poté v testovací verzi postavili na nepájivém poli a ve finální verzi navrhli a objednali vlastní PCB. Pro ochranu naší stanice jsme v open-source programu FreeCad navrhli model krytu, který jsme následně vytiskli na 3D tiskárně. Vymstila se nám však nepřesnost rozměrů krytu a při vkládání stanice do krytu se nám podařilo zlomit SD kartu s operačním systémem, daty a nasazenými programy. Po reinstalaci operačního systému jsme opravili rozměry vytvořeného modelu krytu a vytiskli kryt podruhé, tentokrát ve správných rozměrech.

Hlavní program jsme v prvotní verzi psali ve vysokoúrovňovém programovacím jazyce Python, od kterého jsme nakonec upustili kvůli vysokým technickým nárokům. Naše rozhodnutí nás nakonec dovedlo k jazyku Go vytvořeného Googlem a určeného pro concurrency, rychlou kompilaci i průběh programu, backend webových aplikací a mnoho další funkcionality. Tento jazyk se nám zalíbil natolik že jsme se v něm rozhodli udělat jak API, databázové funkce tak i hlavní program. Program je rozdělen na několik sekvencí z toho hlavní jsou měřící sekvence kde se každou sekundu provádí měření dat ze senzorů a následné odesílání na REST API webové aplikace a channels vytvořené pro komunikaci mezi go rutinami a sekvencí controlleru, která pomocí snímání dat z databáze určuje jestli se jedná o první spuštění PlantHubu, a podle toho určí jestli se spustí inicializační sekvence nebo sekvence zavlažování.

## 2 Hardware

Potřebujeme vyvinout vhodné pouzdro pro naše Raspberry, abychom ochránili citlivé elektronické součástky a ve stejné chvíli měli možnost jednoduše vyjmout stanici z krytu. Budeme ho tisknout na 3D tiskárně pevným PTG filamentem. K zavlažování je potřeba postavit i nádrž na vodu, ze které bude naše čerpadlo přečerpávat vodu k zavlažování, proto jsme se rozhodli pro nádobu s co největším objemem a postavili ji z upraveného 5L soudku.

### 2.1 DHT11

Senzor DHT11 se skládá z jednotky pro měření teploty, jednotky pro měření vlhkosti a převodníku.

Teplotu měří senzor termistorem. Thermistor je keramický polovodič, který zmenšuje svou rezistivitu, když se okolní teplota zvyšuje.

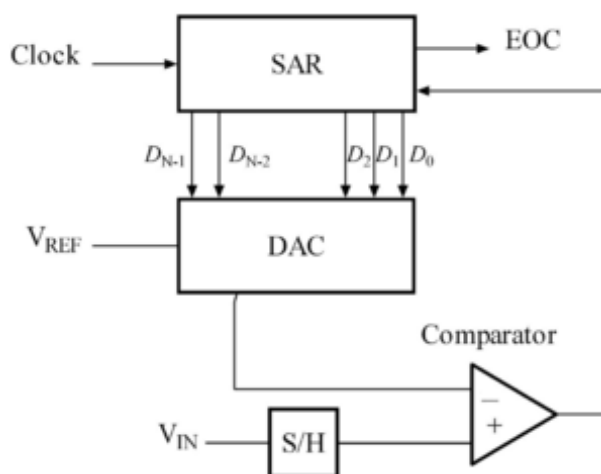
Vlhkost měří senzor na základě rezistivity substrátu umístěného mezi dvěma elektrodami. Tento substrát zachytává vlhkost a vytváří tak vodivé prostředí

### 2.2 Senzor vlhkosti půdy

Jedná se o kapacitní senzor, ten se skládá ze dvou vodivých desek a převodníku. Čidlo funguje na způsob kapacitoru avšak jeho kapacita je ovlivněna vlhkostí, která ovlivňuje dielektrikum mezi dvěma deskami.

### 2.3 MCP3008

RPi má v rámci GPIO pouze digitální vstupy, protože je ale senzor vlhkosti půdy analogový museli jsme použít ADC převodník. Přemapujeme tedy analogový signál do osmi různých digitálních hodnot, kterým se také říká 3bitové rozlišení, to definuje tu nejmenší hodnotu změny, který převodník dokáže rozlišit, této hodnotě se říká LSB (least significant bit). ADC poté pomocí SAR definuje adresu v binárním podání srozumitelným pro digitální vstup.



Obrázek 1: SAR

## **2.4 HC-SR04**

HC-SR04 vydává zvukové vibrace na vysoké frekvenci, neslyšitelné pro lidské ucho. Poté čeká, až se zvuk odrazí zpět, a vypočítá vzdálenost na základě času měřeného od vysílání zvukové vlny k zpětnému přijmutí

Všechny naměřené údaje jsou v převodníku senzoru přepočítány na jednotky dané veličiny a odeslány analogovým signálem do řídicí jednotky.

## **2.5 Čerpadlo**

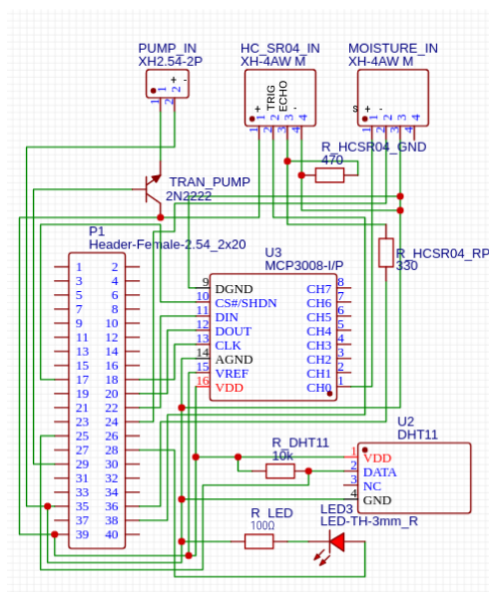
Naše zvolené čerpadlo se skládá z DC motoru, na němž je upevněna centrifuga pro čerpání vody a vlastního pouzdra, z kterého vede otvor pro napojení odtokové hadičky. Čerpadlo je připojeno na zdroj napětí 5V a zem.

## **2.6 2N2222**

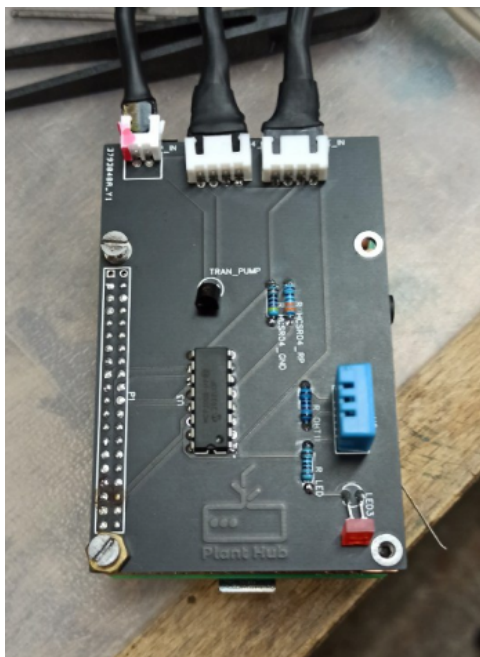
Protože samotný signální pin neposkytuje dostatečné napětí pro chod čerpadla ovládáme jej NPN tranzistorem. Díky našemu vyměnitelnému připojení modulů je možné čerpadlo vyměnit a přívodný kabel používat jako spouštěč čerpadla jakéhokoliv výkonu a nároků na zdroj.

### 3 Obvod

Testovací verzi našeho obvodu jsme postavili na nepájivém kontaktním poli, které jsme používali pouze v testovací verzi. Jakmile jsme měli vše plně odzkoušeno a plně otestováno přešli jsme na profesionálnější řešení. Což tedy v druhé fázi znamenalo sestavit a nechat vytisknout náš vlastní obvod přepracovaného schématu PCB a vytisknutého společností JLCPCB na naše vlastní náklady.



(a) Diagram PCB



(b) PlantHub PCB se senzory

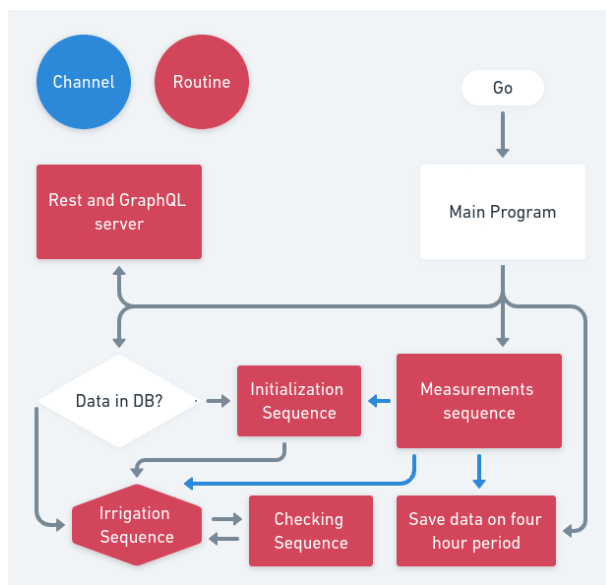
Obrázek 2



## 4 Hlavní program

### 4.1 Postup práce

Náš hlavní program pro zavlažování a komunikaci s databází a WUI jsme začali psát ve vysokoúrovňovém programovacím jazyce Python. Od toho jsme ale nakonec upustili kvůli pomalejšímu průběhu programu, proto jsme přešli na programovací jazyk GO.



Obrázek 3: Vývojový diagram programu

### 4.2 Fáze programu

Fáze programu se spouští buďto v samostatné go rutině a komunikují spolu pomocí channelů nebo na základě podmínek kde se po splnění požadavků ukončí.

#### 4.2.1 Měření

Senzor vlhkosti půdy a DHT11 průběžně posílají naměřená data do Rpi, kde se ukládají do databáze. Jestliže naměřené hodnoty překročí limitní hodnoty, Rpi pošle signál pro otevření tranzistoru což spustí čerpadlo.

#### 4.2.2 Periodické ukládání dat

V samostatné rutině běží funkce pro ukládání naměřených dat v periodě 4 hodin. Naměřená data jsou následně statisticky zobrazena v UI.

#### 4.2.3 Controller

Po spuštění měřicí sekvence a sekvence periodického ukládání dat se spustí buďto inicializační sekvence a nebo zavlažovací sekvence podle toho jestli jsou v databázi data nastavení,

nastavovaná ve WUI. Pokud data nejsou tak program čeká na uložení dat z WUI a LED dioda bliká dvakrát po sobě dokud data nejsou dostupná, pokud ano spustí se zavlažovací sekvence, načtou se limitní data z databáze a v periodě 1 sekundy se budou číst data o vlhkosti půdy.

#### **4.2.4 Inicializace**

Půda musí být ze začátku suchá. Senzor vlhkosti půdy zasuneme co nejhlouběji do půdy. Rpi bude chvíli sbírat data a pak je zprůměruje do hodnoty, která bude sloužit jako limit pro spuštění čerpadla.

V UI jde navíc ještě manuálně nastavit hranice vlhkosti půdy pro spuštění čerpadla.

Nastavit se dá také množství vody, které bude přečerpáno při jednom spuštění a jaká je hranice pro přijatelnou výšku hladiny vody v nádrži. Pokud nejsou tyto hodnoty uvedeny čerpadlo bude vodu přečerpávat, dokud se nezmění hodnota kapacitního čidla pro měření vlhkosti půdy a HC-SR04 použije výchozí nastavení.

#### **4.2.5 Zavlažování**

Čerpadlo začne čerpat vodu a zavlažovat rostlinu. Voda se čerpá tak dlouho, dokud senzor vlhkosti půdy nezmění svou hodnotu nebo dokud není vyčerpán limit přečerpané vody na jedno spuštění.

#### **4.2.6 Kontrola**

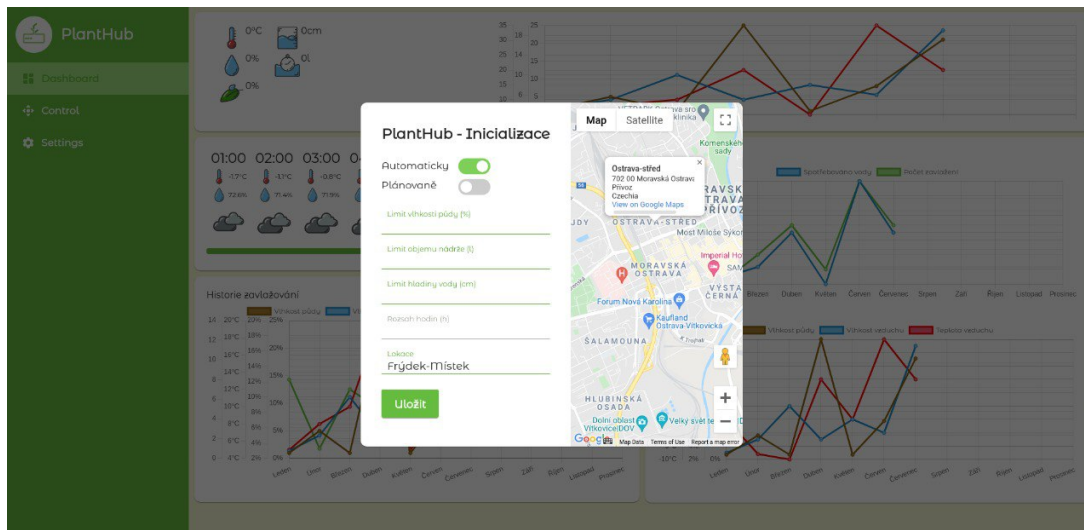
Po ukončení přečerpávání se spustí HC-SR04 a změří výšku hladiny vody. Naměřená data poté odešle do Rpi kde se uloží do databáze. Pokud bude naměřená hodnota nižší, než je limitní hodnota, začne blikat LED dioda a Rpi odešle upozornění o doplnění nádrže do UI. Jakmile bude hladina doplněna, signalizace se vypne.

### **4.3 Ukládání dat**

Náš systém ukládá zvlášť periodicky naměřená data a data naměřená před zavlažováním, dále ukládá nastavení jak pro limity k zavlažování, tak pro WUI.

## 5 WUI

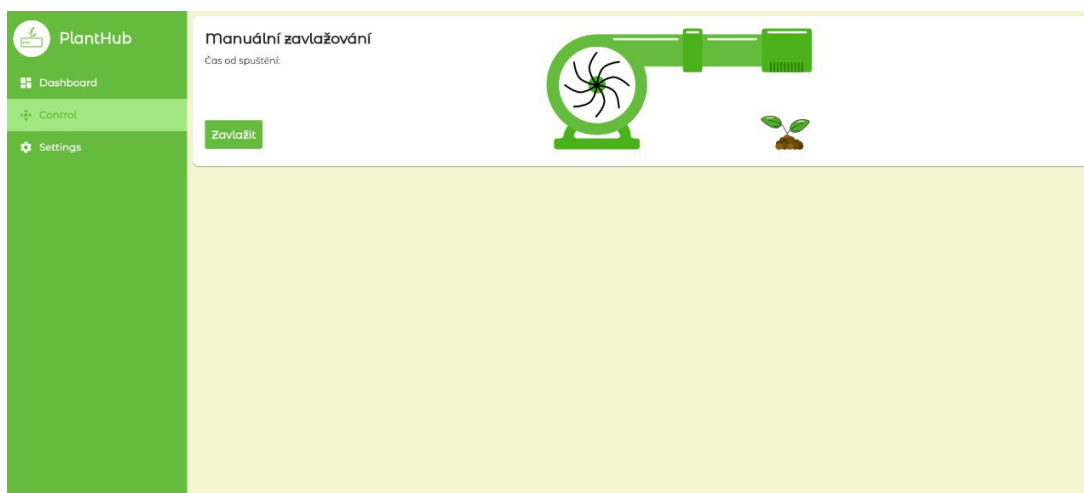
Webovou aplikaci jsme napsali pomocí javascriptového frameworku React.js, CSS frameworku Tailwind a programovacího jazyku Typescript, který je supersetem javascriptu, podporujícím volitelné statické typování. Ve webovém rozhraní je možné zobrazit si statistiky jak živě naměřených dat, tak dat uložených v databázi. Z OpenWeather API získáváme data o předpovědi počasí a do dashboardu renderujeme předpovědi na dalších 15H.



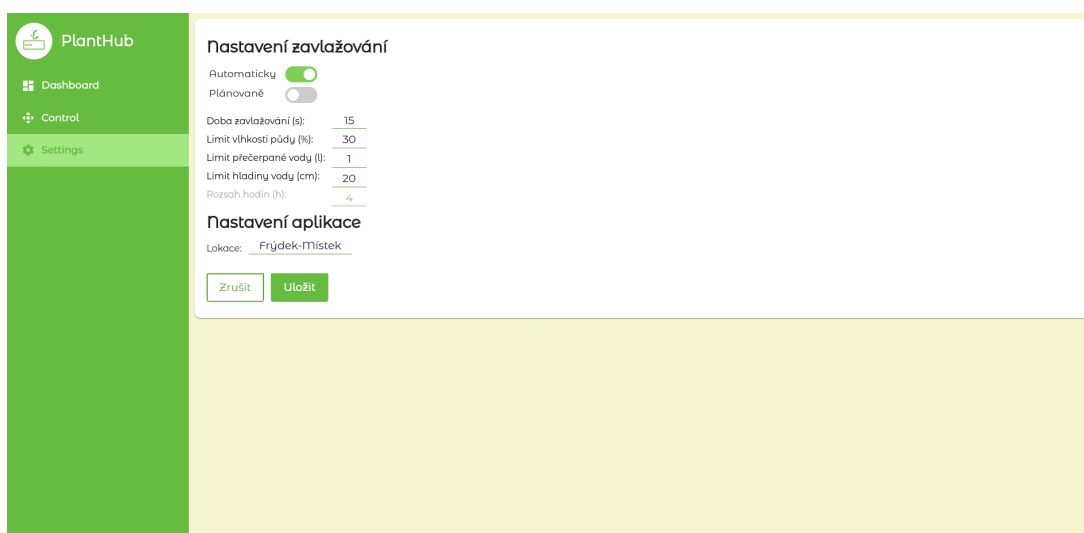
Obrázek 4: Okno prvotního nastavení. Nastaví se zde limity a celkové nastavení aplikace.



Obrázek 5: Dashborad WUI



Obrázek 6: Interaktivní ovládání celého systému lze provést v již zmiňované webové aplikaci. Dovoluje uživateli kdykoliv spustit čerpadlo na zalévání rostliny.



Obrázek 7: V nastavení se dá změnit nastavení aplikace i limitů pro zavlažování. Nastavení se poté uloží do databáze.

## 6 Databáze

Pro databázi jsme se rozhodli použít databázový systém PostgreSQL. Jak vyplývá z názvu, jedná se o SQL databázi, ty jsou vhodné pro ukládání velkého objemu dat, jako právě data z našich senzorů. Webová aplikace používá pro přístup k datům z databáze GraphQL API.

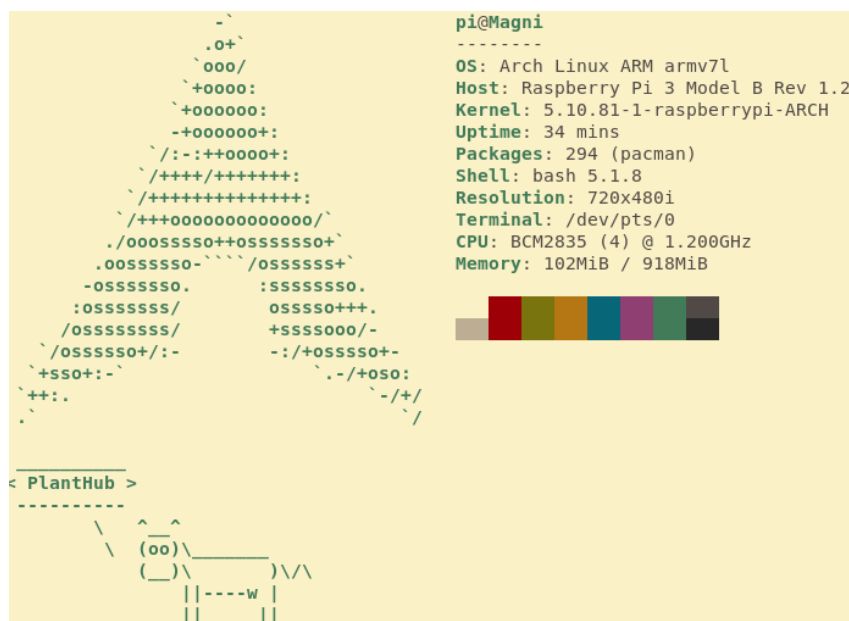
GraphQL je query, který je konkurent řešení REST API. Nabízí efektivnější tvorbu API a také umožňuje efektivnější přístup k datům z databáze. Umožňuje vybrat pouze data, která momentálně aplikace používá a dovoluje vynechat data, která zrovna potřebné nejsou.

## 7 Webový server

Web server jsme napsali v moderním jazyku Go, který nyní roste v popularitě hlavně mezi cloudovými vývojáři a v oblasti vývoje microservices. Dokáže se exekuční rychlostí přiblížit k nízkoúrovňovým jazykům jako je C nebo Rust, ale zároveň zůstává velmi lidsky čitelný a jednoduchý na použití. Na rozdíl od jazyků jako C a Rust má například garbage collector.

## 8 Na čem ještě pracujeme

- RPi case z 3D tisku
- Celková funkcionality hlavního programu a jeho sekvencí
- Realizace nádrže pro vodu



### Obrázek 8: Specifikace RaspberryPi

## 9 Závěr

Filip Sikora je kokor.



## 10 Seznam zkratek a pojmů

- WUI** Webové uživatelské rozhraní
- API** Application programming interface
- REST** Representaion programming interface
- GraphQL** Graph query languge
- PCB** Printed circuit board
- RPi** Raspberry Pi
- ARM** Advanced RISC Machines
- GPIO** General-purpose input/output
- DHT11** Digital humidity temperature (sensor) v.1 1
- HC-SR04** Ultrasonický senzor vzdálenosti
- MCP3008** Analog-digital converter
- Cerpadlo** Ponorné mini čerpadlo eses
- 2N2222** NPN tranzitor
- SPI** Serial peripheral interface
- SAR** Successive-approximation

## 11 Použité nástroje

**Code-OSS** Open-source verze známého textového editoru Visual Studio Code.

**GoLand** Inteligentní IDE vyvinuté společností JetBrains speciálně pro jazyk Go.

**Git** Software pro tracking změn souborů repozitáře projektu uloženého na GitHubu.

**Figma** Webová aplikace pro tvoření vektorové grafiky a návrhů uživatelských rozhraní.

**FreeCad** Open-source program pro modelování 3D objektů.

**EasyEDA** Webová aplikace nabízející jednoduché prostředky pro návrh technických schémat

## 12 Použité zdroje

LaTeX-Tutorial.  
Free Code Camp.  
Go lang.  
React.js.  
Tailwind CSS.  
Typescript.  
Code-OSS.  
Goland.  
Git.  
Figma.  
FreeCad.  
EasyEDA.

## **13 Seznam obrázků**

**Obrázek 1.** SAR

**Obrázek 2. (a)** Diagram PCB

**Obrázek 2. (b)** PlantHub PCB se senzory

**Obrázek 3.** Vývojový diagram projektu

**Obrázek 4.** Okno prvotního nastavení WUI

**Obrázek 5.** Dashboard WUI

**Obrázek 6.** Manuální ovládání z WUI

**Obrázek 7.** Nastavení PlantHubu a WUI

**Obrázek 8.** Specifikace RaspberryPi

## **14 Seznam příloh**