

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra řídicí techniky

Řízení automatického systému hydroponického pěstování rostlin

Václav Truhlařík

Školitel: doc. Ing. Stanislav Vítek, Ph.D.

Obor: Kybernetika a robotika

Květen 2022

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Truhlařík** Jméno: **Václav** Osobní číslo: **492376**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra řídicí techniky**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Řízení automatického systému hydroponického pěstování rostlin

Název bakalářské práce anglicky:

Control of automatic hydroponic plant growing system

Pokyny pro vypracování:

- 1) Proveďte přehled systémů pro hydroponické pěstování rostlin
- 2) Navrhněte dílčí subsystémy pro řízení a monitorování hydroponického systému. Zaměřte se na sledování stavu rostlin, kvality závlahy, osvětlení a uživatelské rozhraní k systému
- 3) Navržené subsystémy implementujte

Seznam doporučené literatury:

- [1] RESH, H. M. A Definitive Guidebook for the Advanced Home Gardener and the Commercial Hydroponic Grower, 551p. 2013.
- [2] SARASWATHI, D., et al. Automation of hydroponics green house farming using IoT. In: 2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA). IEEE, 2018. p. 1-4.
- [3] CHOWDHURY, Muhammad EH, et al. Design, construction and testing of IoT based automated indoor vertical hydroponics farming test-bed in Qatar. Sensors, 2020, 20.19: 5637.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Stanislav Vítek, Ph.D. katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **27.01.2022**

Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce:

do konce letního semestru 2022/2023

doc. Ing. Stanislav Vítek, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Michael Šebek, DrSc.
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Mé poděkování patří doc. Ing. Stanislavu Vítkovi, Ph.D. za odborné vedení, trpělivost a ochotu, kterou mi nejen v průběhu zpracování bakalářské práce věnoval. Dále bych chtěl také poděkovat Ing. Václavu Navrátilovi, Ph.D. za výpomoc při modelování a následném 3D tisku konstrukce LED osvětlení.

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.

V Praze, 20. května 2022

Abstrakt

Tato bakalářská práce ve stručnosti popisuje jak fungují hydroponické systémy a jaké jsou jejich typy. Dále se hlouběji zabývá projektem hydroponické farmy u nás na Fakultě Elektrotechnické, jejím aktuálním stavem, ale hlavně rozšířením z mé strany. To je realizace webového serveru pomocí ESP32 modulu, přidání ESP32-CAM kamerového modulu pro záznam časosběru růstu rostlin a implementace konstrukce umělého osvětlení z 3D tisku s vsazenými RGB LED moduly NeoPixel nad jednotlivými rostlinami.

Klíčová slova: hydroponie, WiFi modul, webový server, LED osvětlení, kamerový modul

Školitel: doc. Ing. Stanislav Vítek, Ph.D.

Abstract

This bachelor thesis briefly describes how hydroponic systems work and what their types are. It also deals more deeply with the project of a hydroponic farm at our Faculty of Electrical Engineering, its current state, but mainly the expansion on my part. Which is, the implementation of a web server using the ESP32 module, the addition of an ESP32-CAM camera module for recording the growth time of plants and the implementation of artificial lighting from 3D printing with embedded RGB LED modules NeoPixel above individual plants.

Keywords: hydroponics, WiFi module, web server, LED lighting, camera module

Title translation: Control of automatic hydroponic plant growing system

Obsah

1 Úvod	1	6.1.3 Ovládací stavy	34
2 Hydroponické systémy	3	6.2 3D tištěné osvětlení	36
2.1 Proč se dělají	3	6.2.1 Hlavní rám	36
2.2 Druhy hydroponických systémů . .	3	6.2.2 Držák na DIN lištu	37
2.2.1 NFT (Nutrient Film Technique)	4	6.2.3 Ramena na LED kroužky . . .	38
2.2.2 EBB & Flow (Flooding and Draining)	4	6.2.4 Propojení a oživení	39
2.2.3 Další příklady hydroponických farem	4	7 Závěr	41
3 Projekt hydroponie u nás na fakultě	7	Literatura	43
3.1 Druh hospodářství	7		
3.2 Použitá automatizace	7		
3.2.1 Hlavní řídicí jednotka	7		
3.2.2 WiFi modul	8		
3.2.3 Senzory pro informace o systému	9		
4 Přeprogramování z Arduino IOT cloud na ESP32 webserver	11		
4.1 Aktuální použití ESP32 s Arduino IOT cloudem	11		
4.2 Webový server pomocí ESP32 . .	11		
4.2.1 Asynchronní webový server . .	11		
4.2.2 SPIFFS	12		
4.2.3 HTML	13		
4.2.4 CSS	14		
4.2.5 JavaScript	14		
4.2.6 RS485	15		
4.2.7 Protokol zpráv	17		
4.3 Webové stránky	18		
4.3.1 Hlavní stránka	18		
4.3.2 Ovládání	19		
4.3.3 pH	21		
5 Implementace kamery	25		
5.1 ESP32-CAM	25		
5.2 Použití	26		
5.2.1 Jeden snímek	27		
5.2.2 Časoběry růstu rostlin	27		
5.2.3 Streamování	27		
6 Umělé osvětlení pro lepší růst	29		
6.1 Přidání Light Emiting Diod (LED)	30		
6.1.1 RGB diody WS2812B	30		
6.1.2 Řízení osvětlení	33		

Obrázky

2.1 Nutrient Film Technique hydroponický systém	4
2.2 Schéma aeroponí farmy	5
3.1 Ovládací tlačítka na řídicí jednotce	8
4.1 Použití <i>Handleru</i> asynchronního webového serveru pro přesměrování na jinou adresu	12
4.2 Struktura SPIFFS v projektu webového serveru	13
4.3 Použití <i>Handleru</i> pro vyvolání souboru ze SPIFFS	13
4.4 Instance tlačítka vygenerovaná pomocí HTML	14
4.5 Instance tlačítka upravená pomocí CSS	14
4.6 Změna textu tlačítka pomocí JavaScript funkce onclick()	15
4.7 Zapojení komunikační linky RS485 pomocí převodníků MAX485E	16
4.8 Zapojení převodníku MAX485E na shieldu desky ESP32	16
4.9 Navigační menu se 4 odkazy ...	20
4.10 Levý FlexBox na stránce ovládání s informacemi týkající se stavového automatu	21
4.11 Element <i>form</i> pro zobrazení a zaznamenání hodnoty jasu LED osvětlení	21
4.12 Webová stránka měření pH s grafem časového průběhu hodnoty pH	23
5.1 Propojení kamerového modulu (vlevo) s <i>USB-UART</i> převodníkem (vpravo)	26
5.2 Webové rozhraní pro ovládání kamerového modulu	26
5.3 Ovládací tlačítka pro jednotlivé funkce kamerového modulu	27
6.1 Absorbční křivka rostlin	29
6.2 Kruhový modul NeoPixel s 12 WS2812B diodami zapojenými do série	30
6.3 Schéma znázorňující konfiguraci pinů pro zapojení WS2812B diody	31
6.4 Schéma znázorňující sériové zapojení WS2812B diod pro řízení po jednom PWM pinu	32
6.5 Ukázky průběhu napětí na PWM pinu	33
6.6 3D model držáku LED osvětlení	36
6.7 3D model hlavního rámu s otvory pro vložení dalších dílů 3D tisku ..	37
6.8 3D model držáku na DIN lištu stažený z internetu	37
6.9 3D model držáku na DIN lištu po mých úpravách	38
6.10 Model hlavního rámu a držáku s detailem na kvádrový výstupek pro stabilitu	38
6.11 Model ramene pro uchycení LED modulu NeoPixel	39
6.12 Přivedení kabelů od 4 NeoPixel modulů z ramen do středu hlavního rámu	40
6.13 Výsledné použití 3D vytisknutého LED osvětlení nad hydroponickou farmou	40

Tabulky

3.1 Stavby stavového automatu	8
4.1 Rozložení <i>FlexBoxů</i> na webové stránce	19
6.1 Stavby stavového automatu týkající se ovládání LED osvětlení	35

Kapitola 1

Úvod

V rámci bakalářské práce navazuji na naši fakultě již rozpracovaný projekt Hydroponické farmy. Takováto farma funguje z hlediska místa a energií mnohonásobně efektivněji, než běžné zemědělství na polích. Zde na fakultě uvádíme do provozu pouze ukázkový model, který dokáže pojmout jen 5 rostlin, nicméně přesahy do reálného využití se začínají čím dál více objevovat po celém světě.

Klasická hydroponická farma dokáže fungovat i bez přebytných technologií jako senzorů, mikropočítačů a řídicích programů, každopádně to pro náš projekt neplatí. V rámci automatizace farmy je hlavní řídicí jednotkou Arduino MEGA 2560 a je vybavena senzorem hodnoty pH. Postupně bude doplňována také o senzory pro pozorování teploty, nebo vodivosti v živném roztoku.

Zároveň mým přínosem pro farmu bude přepracování komunikace po internetu, aby bylo možné farmu pozorovat a ovládat odkudkoliv na internetu. Tuto funkcionalitu zajišťuje ESP32 WiFi modul s webovým serverem a komunikující s řídicí jednotkou Arduina po lince RS485.

Na webu nebudou pouze widgety pro ovládání a pozorování hodnot ze senzorů, ale také odkaz na webové rozhraní ESP32 kamerového modulu. Na webu tohoto modulu bude možné pouštět živě stream z hydroponické farmy, nebo nastavit a nechat ukládat snímky pro časosběr.

V neposlední řadě je mým cílem implementovat umělé osvětlení pomocí RGB LED modulu NeoPixel nad každou rostlinou, pro její největší komfort. Tyto moduly jsou plně nastavitelné a budou vsazeny do mnou vymodelované a na 3D tiskárně vytisknuté konstrukce osvětlení.

Ve výsledku by měl být systém spravovatelný dálkově přes web, automaticky udržovat živé rostliny čerpáním živného roztoku k jejich kořínkům a také zajišťovat adekvátní osvětlení pro rychlejší a stabilnější růst.

Kapitola 2

Hydroponické systémy

2.1 Proč se dělají

S přibývajícím populací se lidé téměř ve všech částech světa začínají potýkat s nedostatkem místa k žití. S tímto problémem je spojen také nedostatek zemědělské půdy pro pěstování plodin. Na uživení jednoho člověka po celý rok je potřeba přibližně 928 m². [1] Najít tolik místa, navíc s výživnou půdou pro pěstování, je velmi obtížný úkol. V neposlední řadě je třeba brát v potaz, že pro vypěstování rostlin jsou z půdy čerpány živiny a naopak jsou do ní vypouštěny všemožné udržovací chemikálie. Půda ničená nepřetržitým užíváním přichází o důležité minerály a může ztratit schopnost zadržování vody. Ve výsledku může na takovýchto místech docházet i k dezertifikaci.

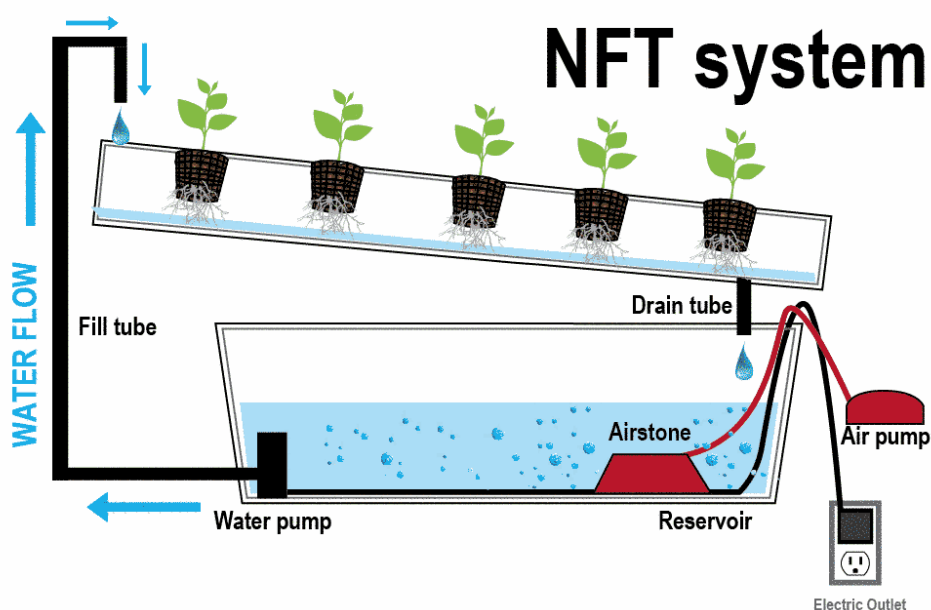
Naštěstí zde přichází na pomoc koncepce hydroponického pěstování plodin. Díky tomu, že většina hydroponických farem nepotřebuje pro své fungování půdu, je velmi jednoduché je stavět do více pater nad sebe, někdy i s dalšími benefity. V porovnání s klasickým zemědělstvím je také velmi jednoduché implementovat takovéto farmy do měst, kde je potřeba potravin nejvyšší. Pro představu 30 patrová hydroponická farma o rozloze 5 akrů je schopna produkovat tolik co normální farma o rozloze 2400 akrů. [2]

2.2 Druhy hydroponických systémů

Hydroponický systém pro pěstování rostlin funguje bez půdy, z které by rostliny získávaly živiny. Proto jsou jim živiny dodávány živným roztokem buď staticky, nebo prouděním. [3] Rostlina je zasazena v médiu jako je například kokosové vlákno, keramzit, rockwool nebo perlit a vsazena do košíku, kterým mohou prorůst kořeny lehce dolů. Na rozdíl od půdy jsou tato média bez živin, proto se musí doplňovat roztokem bohatým na potřebné živiny, jehož základem je voda. Pro správnou výživu je kromě jiného potřeba, aby se u vody v roztoku udržovalo pH a teplota. Z tohoto důvodu jde hydroponické pěstování ruku v ruce s automatizací a používáním senzorů.

2.2.1 NFT (Nutrient Film Technique)

V této variantě hydroponického pěstování rostou kořeny z košíčku rovnou do koryta (trubky), kterým konstantním proudem protéká živný roztok. Neustálé proudění můžeme zajistit vodním čerpadlem, které bude pumpovat roztok do nejvyššího patra (k nejvyššímu bodu jednoho patra), z kterého poteče roztok samospádem do nižších vrstev a zpět do své nádrže. V nádrži roztoku mohou být umístěny senzory pro měření pH, teploty nebo vodivosti vody a také okysličovací pumpa (viz obrázek 2.1).



Obrázek 2.1: Nutrient Film Technique hydroponický systém

2.2.2 EBB & Flow (Flooding and Draining)

Jak naznačuje jméno této metody, hlavní myšlenkou, místo stálého ponoření kořenů do proudu roztoku, je periodické zaplavování vany s kytkami (Flooding) a následný odtok a vysychání (Draining). Zkratka EBB & Flow znamená obdobnou věc, tedy „periodické klesání a stoupání“. [4] Takováto hydroponická farma umožňuje pěstiteli mít plnou kontrolu nad vyživováním rostlin v pravidelných intervalech. S tím přichází zodpovědnost nastavení správného zalévacího cyklu, který může být pro každý druh rostliny zcela odlišný. Svou roli v určení četnosti zálivky nesou i použité substráty. Například keramzit by se měl obecně zalévat 2 až 3 krát za den, zatímco rockwool stačí pouze 1 za den. [5]

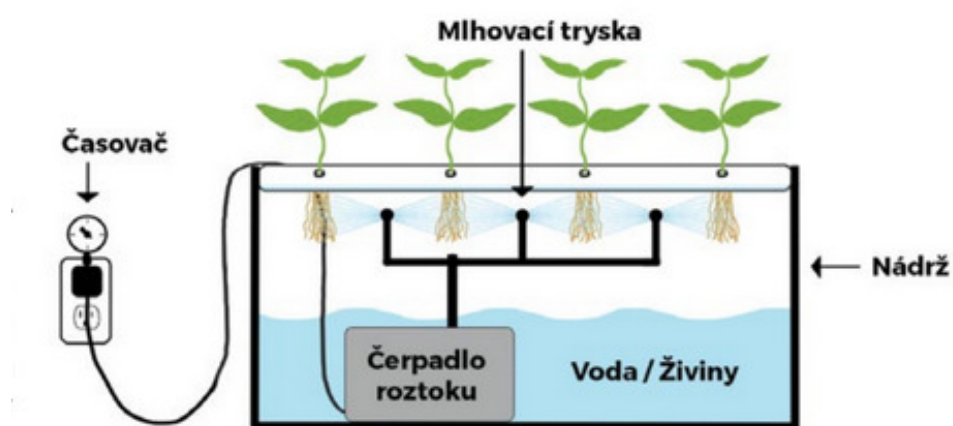
2.2.3 Další příklady hydroponických farem

Jednou z dalších možností je **aeroponie**, kde se rostlinám dodávají živiny nastříkáním živného roztoku přímo na jejich kořínky. Díky tomu je zajištěno

maximální provzdušnění kořenů i absorpce živin. [6] Schéma takové farmy je znázorněno na obrázku 2.2

Obdobou NFT systému je tzv. **Deep Flow Technique** (DFT), ve kterém stojí kořeny rostlin ve vodě nepřetržitě. U takového systému je ale nezbytně důležité zajistit správné dodávání vzduchu. Naopak výhodou oproti ostatním systémům je zajištění přívodu živin i v případě výpadku proudu, nebo jiných poruch s čerpáním.

Nakonec můžeme vzít v úvahu také farmy doplněné o živočichy, respektive ryby. Takovému způsobu pěstování se říká **aquaponie**. Odpad, který vytvářejí vodní živočichové, může být přepracován bakteriemi a použit jako přírodní hnojivo pro rostliny. Z tohoto důvodu je velmi důležitá správná strava ryb, aby bakterie neodumíraly. [7]



Obrázek 2.2: Schéma aeroponí farmy

Kapitola 3

Projekt hydroponie u nás na fakultě

3.1 Druh hospodářství

Budu navazovat na již rozjetý projekt ukázky hydroponie. Naše farma funguje na principu Flood and Drain (2.2.2) a je schopna pojmout 5 rostlin najednou. Dvě vany, respektive dva okruhy, jsou vyrobeny z PVC trubek. Ve vrchním okruhu jsou umístěny rostliny v košíčkách vytištěných na 3D tiskárně. Ve spodní části se budou v budoucnu udržovat správné výživové hodnoty roztoku, jako pH, teplota, vodivost nebo okysličení. Rostliny umístěné v košíčkách jsou obsypány substrátem keramzit, aby v nich stabilně držely.

3.2 Použitá automatizace

3.2.1 Hlavní řídicí jednotka

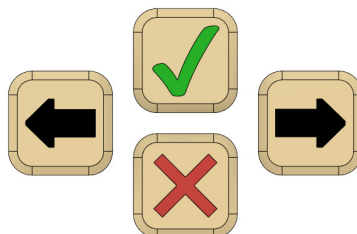
Základnou pro celé řízení systému je deska Arduino MEGA 2560. Obsahuje 54 digitálních a 16 analogových pinů. Pro nahrání kódu disponuje deska 256 kB paměti flash. USB rozhraní pak umožňuje připojení přímo k PC, kde se jeví na virtuálním COM portu číslo 5. Deska nepodporuje komunikaci přes WiFi ani bluetooth[8], proto je k ní připojen také ESP32 modul (3.2.2).

Program řídicí naší farmy je napsaný v C/C++ a je koncipován jako stavový automat. Stavový automat umožňuje strukturovat program do jednotlivých částí, do kterých se systém dostane pomocí výběru funkce *switch/case*. Základní struktura stavového automatu je nastíněna v tabulce 3.1.

Aktuální stav programu a jeho příslušné hodnoty jsou vypisovány na led displej o dvou řádcích na maximálně 16 znaků. Vedle tohoto displeje jsou umístěna také 4 tlačítka (3.1) pro ovládání přístroje, například pro volbu manuální procedury v Menu manuálního spuštění (3.1).

Do stavů označených jako *Automatické* přechází program samovolně po uplynutí stanovené doby. Pro zavlažování hovoříme o době 2 hodin. Během uvádění hydroponické farmy do provozu byla tato hodnota optimalizována z původních 4 hodin, během kterých byly rostliny příliš dlouho bez vody a usychaly. Oproti tomu provzdušňování vody stačí provádět pouze každých 8 hodin. Tato hodnota nemá tak výrazný vliv vzhledem k tomu, že kořeny jsou

po většinu času odhalené a mohou přijímat kyslík ze vzduchu.



Obrázek 3.1: Ovládací tlačítka na řídicí jednotce

Stav	Číslo stavu	Krok	Číslo kroku
Idle	0		
Menu manuálního spuštění	1		
Automatické zavlažování	10	Nastavení časovače Míchání Zavírání ventilu Zavlažování Vypouštění vody Konec cyklu	0 1 2 3 4 5
Automatické provzdušňování	11	Nastavení časovače Spuštění kompresoru Konec cyklu	0 1 2
Manuální zavlažování	50	Stejné kroky jako u automatického procesu	0 - 5
Manuální provzdušňování	51	Stejné kroky jako u automatického procesu	0 - 2
Zobrazení hodnoty pH	100		
Zobrazení naplnění vrchního okruhu	101		

Tabulka 3.1: Stavů stavového automatu

3.2.2 WiFi modul

Jelikož hlavní řídicí deska nedisponuje připojením k internetu, je tato funkce implementována pomocí WiFi modulu, jehož základem je deska DOIT ESP32 devkit V1. ESP32 je deska s dual core procesorem a vestavěnými funkcemi

WiFi a bluetooth. Navíc je vybavena několika rozhraními jako UART, SPI nebo I2C. V naší implementaci je doplněna shieldem, který poskytuje použití sériové komunikace RS485 (4.2.6). Součástí desky shieldu je také obvod Watchdog sloužící pro monitorování běhu programu na ESP32 a jeho případné resetování.

Modul může fungovat v režimu Access Point (AP) nebo Station (STA). V režimu AP si ESP32 vytváří vlastní WiFi, ke které je možné se připojit mobilním zařízením, notebookem, nebo jinými moduly a senzory. Naopak v režimu STA se modul připojuje k již existující WiFi síti. Provedení STA je právě implementováno v našem systému a umožňuje tedy jeho připojení k internetu.[9]

Původní implementace modulu byla myšlena tak, aby komunikovala data na Arduino IoT Cloud. Na cloudu by se data ukládala a zobrazovala pro vzdáleného uživatele. Arduino IoT cloud je platforma, na které si uživatel může poskládat vlastní grafické rozhraní z již před-připravených widgetů. [10]

■ 3.2.3 Senzory pro informace o systému

Jelikož nepěstujeme rostliny v půdě, z které by získávaly živiny, musí se jim dostávat nejlepší možné výživy ve formě živného roztoku. Takovýto roztok, jak už bylo řečeno výše, je v základu voda s určitými vlastnostmi. Pro co nejlepší pohodlí rostlin by voda měla mít určitou teplotu, pH a vodivost.

Hodnota pH živného roztoku ovlivňuje vstřebávání živin rostlinami. Například kyselé roztoky podporují vstřebávání hliníku, vodíku a manganu, jejich přebytek v rostlině může být nežádoucí. Protože se budeme snažit udržovat pH roztoku v rozumných mezích (ne velmi vysoké ani nízké), stačí nám levnější laboratorní elektrochemický pH senzor.

Další v tuto chvíli použitý senzor je hladinový senzor pro udávání informace o naplnění horního okruhu. Tento senzor se využívá při napouštění vody do horního okruhu a kontroluje se jím, aby voda nepřetekla. Senzor udává hodnoty Naplněný, Nenaplněný.

Momentálně se nijak dále npracuje s hodnotami ze senzorů než, že se dají zobrazit uživateli. Nicméně v budoucnu se zcela jistě budeme věnovat oživení dávkování pH+ a pH-, přidání měření teploty a topení, či dávkování hnojiva, abychom udržovali hodnoty živného roztoku v žádaných mezích. [10]

server dokáže velmi rychle zpracovávat neuspořádaně přicházející dotazy z prohlížeče, který je na něj připojen, a postupně na ně odpovídat. Jelikož se jedná právě o asynchronní server, neběží jeho kód na hlavní smyčce, ale pouze se nastaví jak má server odpovídat na jednotlivé dotazy. Toto nastavení probíhá v rámci části kódu `setup()` a je vykonáno jen jednou po restartu serveru.

Server neustále poslouchá pro přicházející požadavky od klientů (prohlížečů). V případě, že se objeví spojení, nastane cyklus odpovídání:

1. Je přijato TCP připojení.
2. Připojení je zabaleno jako *Request* objekt (požadavek).
3. Server zpracuje příchozí *Rewrites* (přepisují url požadavku a přidávají případné *get* parametry).
4. Postupně projde všechny připojené *Handlers* (vykonávají přesné akce určené pro konkrétní požadavky), vybere ten správný (nebo defaultní) a připojí jej.
5. Provede příkazy v rámci zvoleného *Handler*.
6. Následně připojí *Response* objekt (odpověď na požadavek).
7. Po odeslání *Response*, se uzavře *Request* a je uvolněna použitá paměť.

[11]

Na obrázku 4.1 můžeme vidět příklad *Handleru*, který zpracovává požadavek na url `/camera` pomocí metody `HTTP_GET`. Požadavku dále přiřazuje funkcionality `redirect(http://147.32.223.210)`, která prohlížeč, který požadavek vyslal, přesměruje danou url adresu.

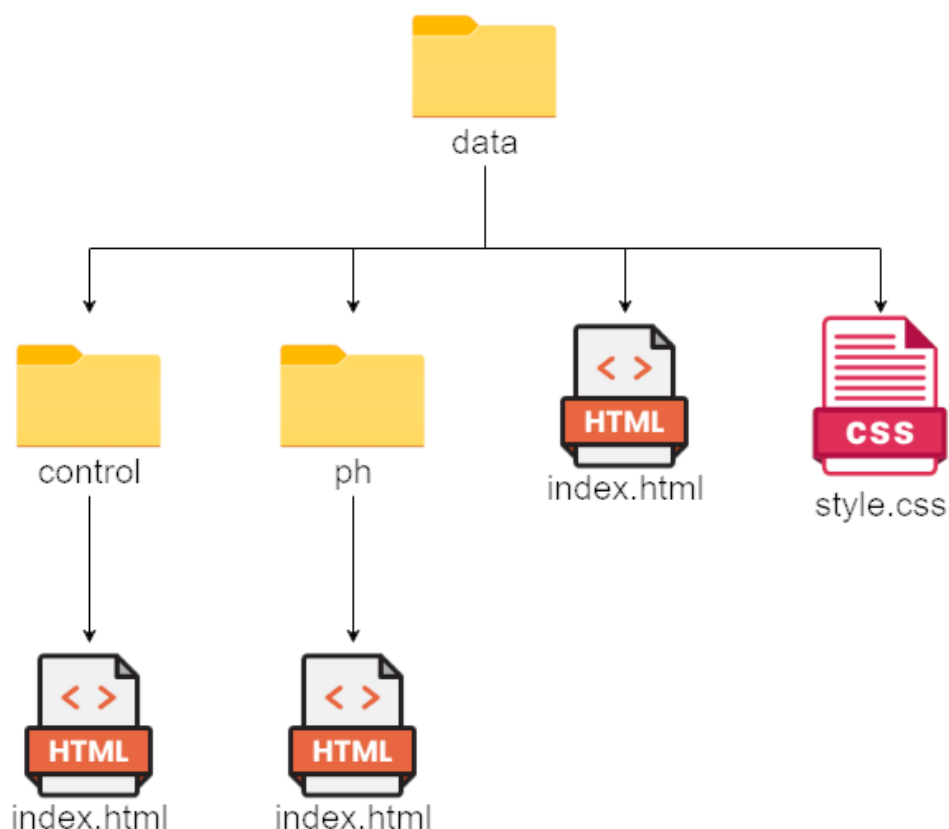
```
server.on("/camera", HTTP_GET, [])(AsyncWebServerRequest *request){
  request->redirect("http://147.32.223.210");
};
```

Obrázek 4.1: Použití *Handleru* asynchronního webového serveru pro přesměrování na jinou adresu

4.2.2 SPIFFS

Serial Peripheral Interface Flash File System, zkráceně SPIFFS, je souborový systém pro flash zařízení na vestavných systémech. SPIFFS podporuje *wear levelling* (prevence proti opotřebení), nebo prověřování stavu souborového systému. Pro naše použití poslouží SPIFFS jako malá SD karta (v základu o velikosti 1,5 MB), na kterou uložíme `.html` a `.css` soubory pro zobrazování na web. [12]

Jelikož se jedná o souborový systém, je možné v něm vytvářet jednotlivé soubory, adresáře, podadresáře a další. Na obrázku 4.2 je vidět struktura souborového systému použitého v našem projektu. [13]



Obrázek 4.2: Struktura SPIFFS v projektu webového serveru

Z těchto adresářů se soubory posílají jako text v odpovědi na HTTP dotaz z prohlížeče. Na následujícím obrázku 4.3 je vidět trochu upravený *Handler* pro posílání *.html* souboru ze SPIFFS do prohlížeče.

```

server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
  request->send(SPIFFS, "/index.html");
});
  
```

Obrázek 4.3: Použití *Handleru* pro vyvolání souboru ze SPIFFS

4.2.3 HTML

Hyper Text Markup Language neboli HTML je programovací jazyk pro vytváření webových stránek, v kterém se kód skládá použitím množiny značek (tagů). Jednotlivé tagy a jejich vlastnosti jsou uzavřeny úhlovými závorkami < a >. Tagy jsou většinou používány v párech, kdy koncový tag je totožný s otevíracím, pouze má před názvem znak lomítka. Část kódu nebo textu obklopena otevíracím a uzavíracím tagem se nazývá element a většinou se projeví jako hotový prvek na webové stránce. Například element tlačítka

definovaný jako `<button>Stiskni!</button>` se projeví na stránce jako hotové tlačítko jako na obrázku 4.4. [14]



Obrázek 4.4: Instance tlačítka vygenerovaná pomocí HTML

■ 4.2.4 CSS

Pokud bychom používali k tvorbě webových stránek pouze jazyk HTML, byly by všechny stránky velmi podobné a vzhledově nezajímavé. Styl webu je naštěstí možné velmi dobře upravovat pomocí jazyka CSS. *Cascading Style Sheets*, tedy zkráceně CSS, je jazyk jehož hlavním účelem je doplnit již hotovou HTML stránku o styl zobrazení. V rámci takovéto úpravy stylu je možné měnit rozložení stránky, velikosti elementů, barvy, fonty a další. Nový styl pro webovou stránku můžeme založit buďto přímo v kódu HTML, kam do hlavičky přidáme tagy `<style></style>`, nebo je možné vytvořit úplně nový dokument. Výhodou použití nového dokumentu je velmi jednoduchá distribuce na více stránek najednou.

Další možností, jak měnit styl jednotlivých prvků, je přidat atribut `style=""` přímo do otevíracího tagu elementu. Pro ukázkou, přidáním `style="background-color: #659DBD; color: #FBEEC1; border: outset 3px #4c7288; padding: 5px 10px;"` do otevíracího tagu tlačítka výše, se tlačítko změnilo na výrazně zajímavější (viz obrázek 4.5). [15]



Obrázek 4.5: Instance tlačítka upravená pomocí CSS

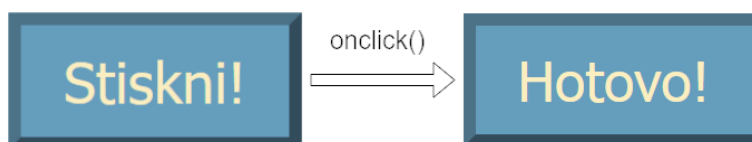
■ 4.2.5 JavaScript

Třetí jazyk pro úpravu webových stránek, *JavaScript* (JS), je ten, který stránky přivádí k životu. Bez použití JavaScriptu by se každá stránka vygenerovala vždy jednou a žádnou interakcí člověka by se nedala změnit (pouze přepnout na jinou stránku, což umožňuje HTML). Jazykem JavaScript tedy

doplňujeme psaný v kombinaci HTML a CSS, a zároveň s ním můžeme měnit většinu atributů pocházejících z těchto dvou jazyků.

Pomocí JavaScriptu je možné přepsat text uvnitř elementu, změnit jeho barvu, styl zobrazení nebo ho nechat úplně zmizet. Abychom mohli JS použít, musíme ho podobně jako CSS vložit do HTML kódu. To můžeme opět udělat několika způsoby. Základní možností je přidání `<script></script>` tagů, do kterých se zapíše jednotlivé funkce JavaScriptu. Pokud chceme stejné funkce používat na více stránkách, je možné vytvořit vlastní dokument obsahující právě tyto scripty a na ten se odkazovat z HTML kódů jednotlivých stránek.

A v neposlední řadě je opět možné přiřadit JS funkci elementu přímo v jeho otevíracím tagu. Rozšířením našeho tlačítka z předchozí podsekcce 4.5 o funkcionalitu `onclick=this.innerHTML = "Hotovo!"` můžeme jeho stisknutím změnit nápis ze *Stiskni!* na *Hotovo!*, viz obrázek 4.6. [16]



Obrázek 4.6: Změna textu tlačítka pomocí JavaScript funkce `onclick()`

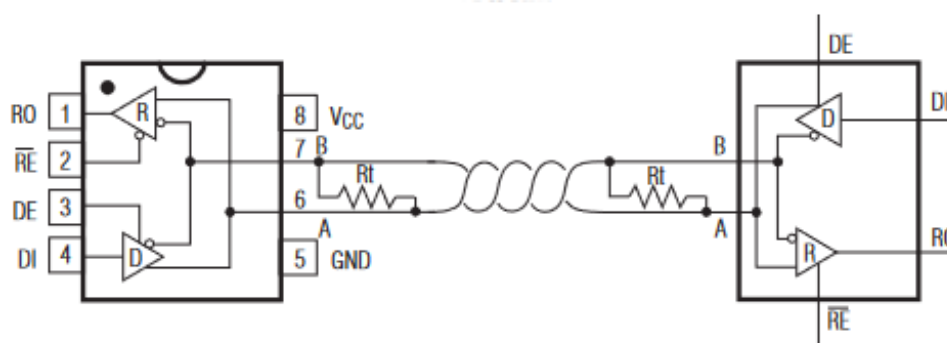
4.2.6 RS485

Aby informace poskytované serverem na webový prohlížeč byly pravdivé, je potřeba komunikovat příslušné stavy hodnoty z řídicí jednotky na ESP modul serveru. Tato komunikace je zprostředkována pomocí přenosové linky RS485.

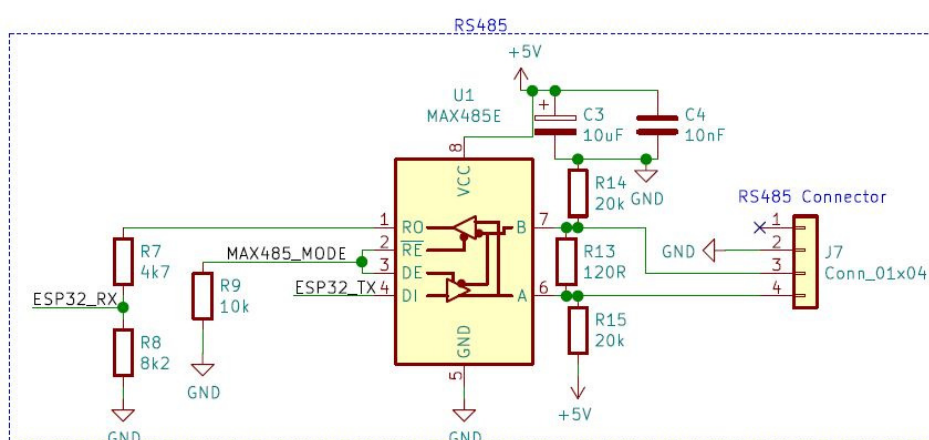
Tato linka je typická komunikací po jednom páru vodičů, na kterém může probíhat komunikace v obou směrech. Vodiče jsou vysílačem buzeny v protifázi, zatímco přijímač rozeznává jejich diferenciální napětí a podle toho určuje logickou hodnotu přenosu. Použitím krouceného páru vodičů pak také zamezíme aditivnímu rušení, které může po cestě nastat. Linka RS485 pracuje s rozdílovým napětím okolo 2 V oproti lince RS232, která typicky pracuje s napětovými úrovněmi -12 V a +12 V. [17] Typické zapojení vysílače a přijímače spojeného linkou RS485 je vidět na obrázku 4.7, kde je sprostředkováno převodníky MAX485E.

Jelikož ani jedna z desek Arduino MEGA, ESP32, nedisponuje přímo rozhraním pro komunikaci po lince RS485, jsou jejich shieldy vybaveny převodníky MAX485E. Tento převodník je zapojen na obou stranách velmi podobně dle schéma na obrázku 4.8. Důležité je si povšimnout označení vyvedených pinů vlevo, kde máme *Receiver Outpu* (přijímač) vyvedený k ESP32 na pin označený jako *ESP32_RX* (v naší konfiguraci pin 16), dále pak *Drive Input* (vysílač) vyvedený na pin *ESP32_TX* (pin 17) a nakonec spojené piny *Receiver output Enable* a *Drive input Enable* (nepovolení přijímání a povolení odesílání), které jsou přivedeny na jeden pin *MAX485_MODE* (pin 4).

Jak je naznačeno v odstavci výše, pin *ESP32_RX* funguje jako přijímač



Obrázek 4.7: Zapojení komunikační linky RS485 pomocí převodníků MAX485E



Obrázek 4.8: Zapojení převodníku MAX485E na shieldu desky ESP32

pro data přicházející z řídicí jednotky. Naopak pin *ESP32_TX* má funkci vysílače a skrze něj se posílají data z ESP32 do jednotky Arduino MEGA. Povolovací piny, díky převrácené polaritě, mohou být spojeny dohromady a být ovládány pouze jedním přepínacím pinem z ESP. Ve chvíli, kdy je hodnota pinu *MAX485_MODE* rovna logické 0, je povoleno přijímání a odesílání zakázáno. V logické hodnotě 1 pinu *MAX485_MODE* jsou funkcionality prohozeny. Na schématu si můžeme všimnout také *pull-down* rezistoru *R9*, který zajišťuje, že převodník je ve výchozí pozici nastaven na přijímání.

Posílání a přijímání dat přes RS485 je řešeno pomocí knihovny *SoftwareSerial*. Knihovnu jsme dále doplnili knihovnou *RS485_protocol*, která zajišťuje posílání celých zpráv typu *byte[]* (pole bajtů). Celý proces posílání zprávy pomocí knihovny *RS485_protocol* je následující:

1. Pošle se indikátor *STX* začátku zprávy.
2. Všechny datové bajty zprávy se pošlou zdvojeně a invertovaně pro zkontrolování správnosti.
3. Pošle se ukončovací indikátor *EXT*.

4. Nakonec se pošle kontrolní součet typu 8 bitového CRC.

Přijímání zprávy funguje obdobným způsobem, kdy se aktivně čeká na příchozí data, která se následně kontrolují, čtou a řadí podle příjmu *STX* a *ETX*. Výsledkem správně přečteného proudu dat je opět *byte[]*. Pokud nedorazí žádná data v rámci vyhrazeného časového okna (v základu 500 ms), vrací se program zpět do hlavní části, kde přestal.

4.2.7 Protokol zpráv

Pro ulehčení komunikace mezi řídicí jednotkou Arduino MEGA a WiFi modulem ESP32 se posílá jen několik zpráv. Každých 5 sekund jsou komunikovány zprávy z řídicí jednotky do ESP, na které je odpovězeno zprávou s potenciálními daty z ESP, respektive z webu.

Struktura jednotlivých zpráv je velmi jednoduchá, kdy na začátku posíláme jméno příjemce a odesílatele. Typ zprávy z řídicí jednotky je tedy adresován následující sekvencí na začátku zprávy: *"ESPMEGA"*. Za adresami následuje vždy bajt určující datový typ zprávy. Pro zprávu týkající se procesních stavů řídicí jednotky se užívá datový typ označený *"0"*. Oproti tomu zpráva nesoucí informace o LED osvětlení je značena datovým typem *"5"*.

Pro zprávu ze strany řídicí jednotky, která nese informace o stavu programu, je struktura následující:

1. 3 bajty číselně značící aktuální stav dle tabulky 3.1.
2. 1 bajt číselně označující aktuální krok stavu, opět dle tabulky 3.1.
3. Oddělovací znak *"x"*.
4. 4 bajty nesoucí číselnou hodnotu pH ve formě: desítky, jednotky, desetiny, setiny.
5. Oddělovací znak *"x"*.
6. 1 bajt s číselnou hodnotou *"0"* nebo *"1"* udávající, zda je naplněn horní okruh vodou.

V ESP se tedy zpráva ve tvaru *"ESPMEGA0101x0796x0"* interpretuje jako:

1. Příjemcem je ESP.
2. Odesílatelem je Arduino MEGA.
3. Aktuální stav dle tabulky 3.1 je *Automatické zavlažování* a právě se nachází v kroku *Míchání*.
4. Poslední zaznamenaná hodnota pH je 7.96.
5. Horní okruh není plně napuštěn.

Zpráva datového typu "5" je poskládána podobným způsobem, ale posílají se v ní informace týkající se LED osvětlení tedy: jas, hodnota červené, hodnota zelené a hodnota modré barvy. Zprávy přicházející od ESP32 do řídicí jednotky jsou velmi podobné, pouze se posílají jen když je vyžadována změna uživatelem z webových stránek.

4.3 Webové stránky

Dle předchozí části je jasné, že webové stránky jsou nakonfigurovány pomocí 3 jazyků. HTML vytvářející elementy zobrazené na stránce, CSS upravující jejich vzhled a vzhled celých stránek a JavaScript zajišťující aktivní funkce stránky jako přepisování hodnot a zadávání požadavků uživatele.

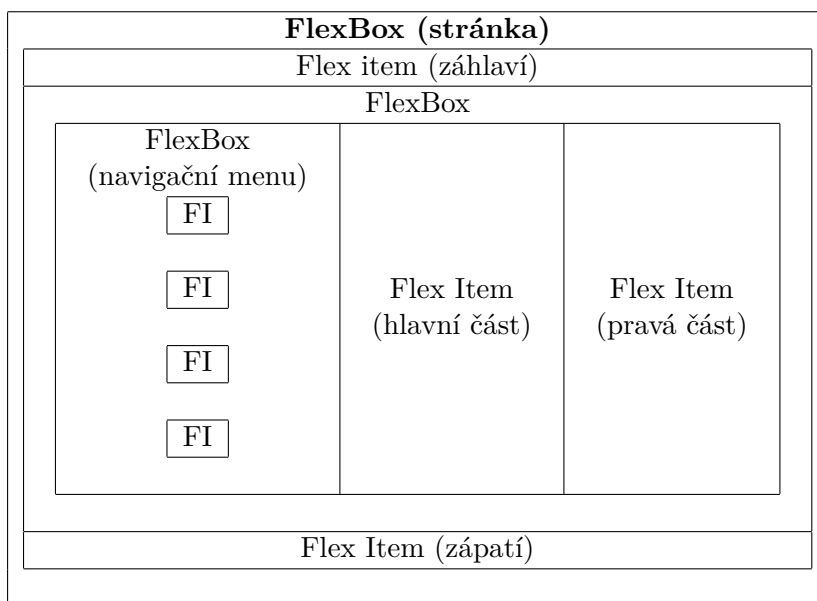
Na rozdíl od Arduino IoT Cloudu je výhoda vlastního serveru v tom, že můžeme mít hned několik různých webových stránek. Každá stránka má svůj vlastní HTML kód a zobrazuje rozdílné věci. Jelikož jsme ale chtěli zachovat jednotný vzhled všech stránek, je použit jeden CSS soubor *style.css*. Poté je nutné tento soubor vložit jako link do HTML kódů, aby prohlížeč věděl, že má soubor *style.css* hledat a kde. U jednotlivých elementů jsou jim jejich CSS vlastnosti přiřazeny pomocí atributu *class=""* v otevíracím tagu. Proto velmi nepřehledný a dlouhý atribut *style=""* (viz 4.2.4) nahrazujeme pouze indikativním názvem v atributu *class="nav_button"*.

Všechny 3 stránky mají velmi podobné, téměř totožné rozložení, což je dáno také tím, že pro ně používáme již zmíněný jeden *style.css* soubor. Přesto můžeme rozložení jasně definovat na 5 základních částí. Jednotlivé části jsou uspořádány pomocí CSS *style* atributu *display: flex*, který nastaví zobrazování daného elementu jako *FlexBox*. *FlexBox* je forma zobrazování, která se velmi dobře přizpůsobuje obsahu, nebo velikosti stránek. Element, který má nastavený tento styl zobrazení funguje následně jako kontejner, do kterého můžeme vkládat další elementy jako *Flex* itemy. Jak jsem již zmínil, je jednoduché nastavit, aby se itemy v kontejneru zobrazovaly například od levého nebo pravého kraje, aby měly mezi sebou rozestupy nebo byly namačkány, nebo zda mají mít velikost podle svého obsahu, či se roztáhnout a zaplnit celý prostor jejich rodičovského kontejneru. Zároveň je možné do kontejnerů vkládat další kontejnery. Tabulka 4.1 nastiňuje rozložení stránky pomocí *FlexBoxů*.

Celkem jsou na webu nyní implementovány 3 stránky, *Hlavní stránka*, *Ovládání* a *pH*. Odkazy na tyto stránky, společně s odkazem na webové rozhraní ESP32 kamery 5, jsou umístěny vždy vlevo na stránce v navigačním menu (obrázek 4.9). Kliknutím na odkaz se vytvoří *HTTP_GET* instance, kterou následně zpracuje asynchronní webový server a přesměruje nás na požadovanou stránku.

4.3.1 Hlavní stránka

Hlavní stránka je koncipována pouze jako uvítací obrazovka, která se vyvolá při zadání IP adresy serveru (<http://147.32.223.214>) do vyhledávače. Na



Tabulka 4.1: Rozložení *FlexBoxů* na webové stránce

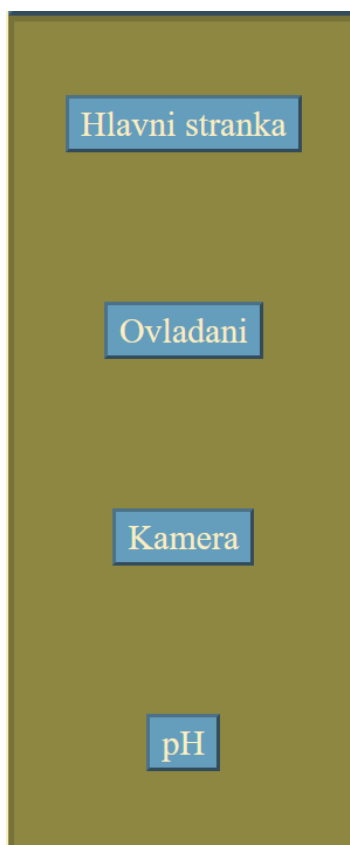
stránce se uživateli zobrazí obrázek modelu naší hydroponické farmy a vlevo pomocí navigačního menu je možné překliknout na další stránky serveru. V záhlaví stránky (dle rozložení v tabulce 4.1) se vlevo nachází název webu *Hydroponi FEL* a vpravo je zobrazeno logo univerzity. V zápatí stránky je v pravém rohu umístěn ještě samostatný znak lva pocházející z loga školy. Zobrazení záhlaví, zápatí a navigačního menu je na všech 3 stránkách totožné (viz obrázek 4.12).

■ 4.3.2 Ovládání

Dalo by se říci, že stránka Ovládání je skutečná hlavní stránka webového serveru, jelikož skrze ní je možné komunikovat s řídicí jednotkou. Stránka se uživateli zobrazí v prohlížeči kliknutím na odkaz *Ovládání* v navigačním menu, nebo zadáním její url přípony */control* za IP adresu serveru ve vyhledávacím řádku.

Oproti hlavní stránce má stránka ovládání v hlavní části dva FlexBoxy. V levém FlexBoxu se nachází informace týkající se stavového automatu, zobrazování stavů a jejich ovládání pomocí tlačítek (viz obrázek 4.10). Je zde vyobrazena obdoba LCD displeje, který je fyzicky na řídicí jednotce, na němž se vypisují stavy a jejich informace podobně jako na jeho skutečné předloze. Pod elementem displeje jsou umístěna 4 ovládací tlačítka se stejnou funkcionalitou a podobným vzhledem, jako jsou fyzická tlačítka na řídicí jednotce. A nakonec jsou zde umístěna tlačítka pro manuální spuštění provzdušňování a manuální spuštění zavlažování.

V pravé hlavní části stránky jsou pak umístěny informace týkající se LED osvětlení 6.1. Mezi důležité hodnoty osvětlení patří *jas (brightness)*, *červená (red)*, *zelená (green)* a *modrá (blue)* složka světla. Každá z těchto hodnot je



Obrázek 4.9: Navigační menu se 4 odkazy

typu *uint8_t* tedy 8 bitový *unsigned integer* a může se pohybovat v rozmezí 0 až 255. Pro zobrazení a zároveň zaznamenání vstupů od uživatele jsem zvolil element `<form></form>`. Tento element má funkci kontejneru pro různé typy vstupních elementů, jako jsou vstupní textová pole, nebo potvrzovací tlačítka. [18] Princip zobrazení je následující:

1. Jméno hodnoty, která je v elementu *form* zobrazována.
2. Tučně vypsaná aktuální hodnota přijímaná po komunikaci z řídicí jednotky.
3. Textový `<input>` element, do kterého uživatel zapíše novou požadovanou hodnotu.
4. *Submit* (potvrdit) tlačítko, kterým se přečte hodnota z `<input>` elementu a pošle se do řídicí jednotky k zapsání.

Elementu *form* je třeba také vyplnit atribut `action="/get"`, který je zpracován pomocí *handleru* asynchronního serveru, a podle parametru hodnoty je s ním následně naloženo. Na obrázku 4.11 je vidět již vygenerovaný element pro hodnotu jasů LED osvětlení.



Obrázek 4.10: Levý FlexBox na stránce ovládání s informacemi týkající se stavového automatu



Obrázek 4.11: Element *form* pro zobrazení a zaznamenání hodnoty jasu LED osvětlení

■ 4.3.3 pH

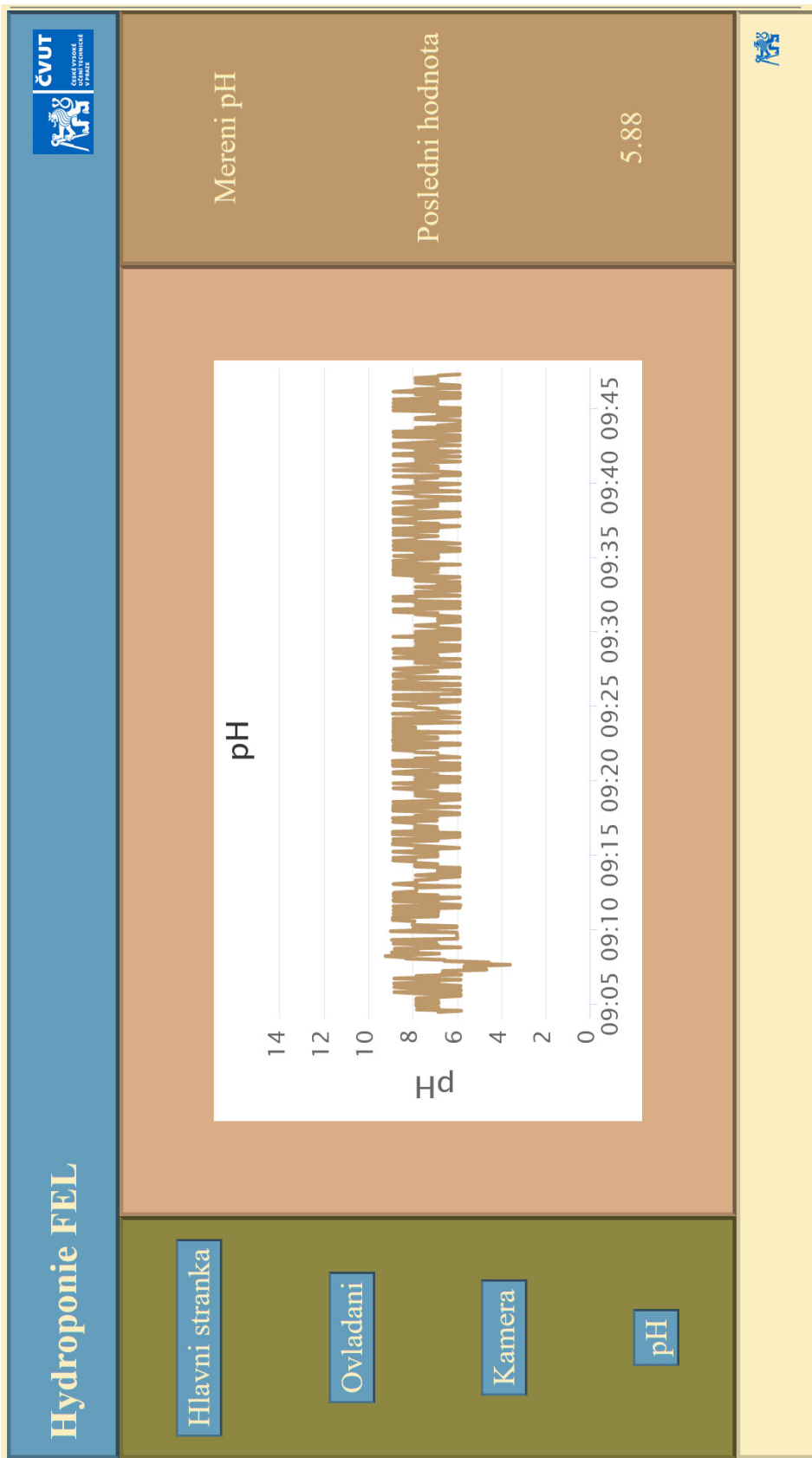
Na třetí stránce běžící na webovém serveru je zobrazeno měření pH živného roztoku ve spodním okruhu naší hydroponické farmy. V hlavní části stránky se pak nachází graf implementovaný pomocí knihovny *highcharts*. Knihovna umožňuje pomocí JavaScriptu přepisovat HTML element `<div></div>`, aby se zobrazoval jako lineární graf. V rámci JS se dal také změnit styl zobrazování jednotlivých prvků grafu jako například velikost, barvu písma a samotný text titulku grafu nebo názvů os, změnit minimální a maximální hodnoty na ose *y*, nebo změnit barvu a tloušťku datové křivky.

Jak již bylo zmíněno, stránka je doplněna o JS funkce, které zajišťují vykreslování grafu. Každých 5 sekund je vygenerován *HTTP_GET* požadavek pro získání aktuální hodnoty pH. Tato hodnota se následně spojí s aktuální hodnotou času a následně je tato dvojice vepsána jako bod o souřadnicích [*x*,

y] do grafu. Ve chvíli kdy je v grafu více než jeden bod, propojí se a zobrazí křivku časového průběhu hodnoty pH. Maximální počet bodů zobrazených v grafu je nastaven na 1440, tedy vynásobením periodou přidávání bodů (5 s), získáme maximální časový průběh pH hodnoty za posledních 7200 sekund, neboli 2 hodiny.

Na rozdíl od hlavní stránky nebo stránky pro ovládání, je na této konkrétní stránce upravena také pravá část stránky (viz rozložení 4.1). U předchozích dvou stránek byl v pravé části zobrazován pouze název aktuální stránky, tedy *Hlavní stránka*, nebo *Ovládání*. Zde je kromě názvu *Měření pH* zobrazována také poslední hodnota komunikovaná z řídicí jednotky, která zároveň odpovídá poslednímu bodu v grafu. Tato hodnota je aktualizována stejně jako graf každých 5 sekund.

Snímek ze stránky pro zobrazení měření pH je vidět na obrázku 4.12. Pro ilustraci je hodnota pH zašumělá, aby byla znatelná funkcionalita grafu. Jinak se hodnota pH moc nemění a drží se kolem 8, což je odpovídající pro pitnou vodu, která teče na fakultě. Zároveň zde vidíme skutečné rozložení stránek, název webu a logo v záhlaví, navigační menu vlevo, hlavní část uprostřed, pravou část a také znak v pravém rohu zápatí stránky.



Obrázek 4.12: Webová stránka měření pH s grafem časového průběhu hodnoty pH

Kapitola 5

Implementace kamery

Dalším prvkem doplněným na našem projektu hydroponické farmy je kamerový modul, který bude zaznamenávat obraz farmy. Jako kameru jsem zvolil modul ESP32-CAM, který byl pro tento projekt hydroponické farmy zvolen již během předchozích prací na projektu.

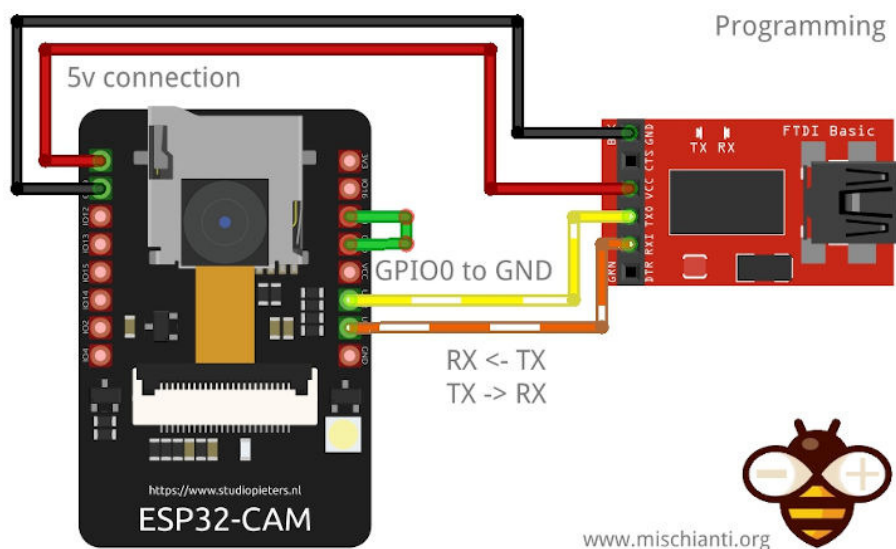
5.1 ESP32-CAM

Jak naznačuje název, jedná se o spojení ESP32 WiFi modulu s kamerovým modulem OV2640 s rozlišením 2 MP. Součástí modulu je také čtečka paměťových karet typu *microSD* pro ukládání snímků a zároveň přídavný *PSRAM* obvod kvůli zpracování velkého objemu dat v programu. Navíc je modul vybaven bílou přisvětlovací LED diodou, která je od výroby nekorektně připájena a svítí po celou dobu běhu programu a zároveň problikne vždy, když dojde k zápisu na SD kartu. Na druhou stranu tato chyba nijak neovlivňuje běh programu a naopak je dobrou indikací, že je ke kameře přivedeno napájení, které je na úrovni + 5 V.

Celkově se jedná o velmi malý modul, na kterém už je přiděláno několik věcí, jako právě zmíněná čtečka *microSD* karet, a tudíž nejspíš nezbylo místo pro *microUSB* port pro připojení k PC. Aby bylo možné desku naprogramovat, je třeba použít libovolný *USB-UART* převodník. Korektním propojením jednotlivých pinů převodníku a kamerového modulu (jako na obrázku 5.1) je možné kameru následně programovat jak jsme zvyklí u ESP32 s *microUSB* portem. Při samotném nahrávání je ale třeba přepínat modul do *bootovacího* režimu propojením pinů *IO0* a *GND* na kamerovém modulu. [19]

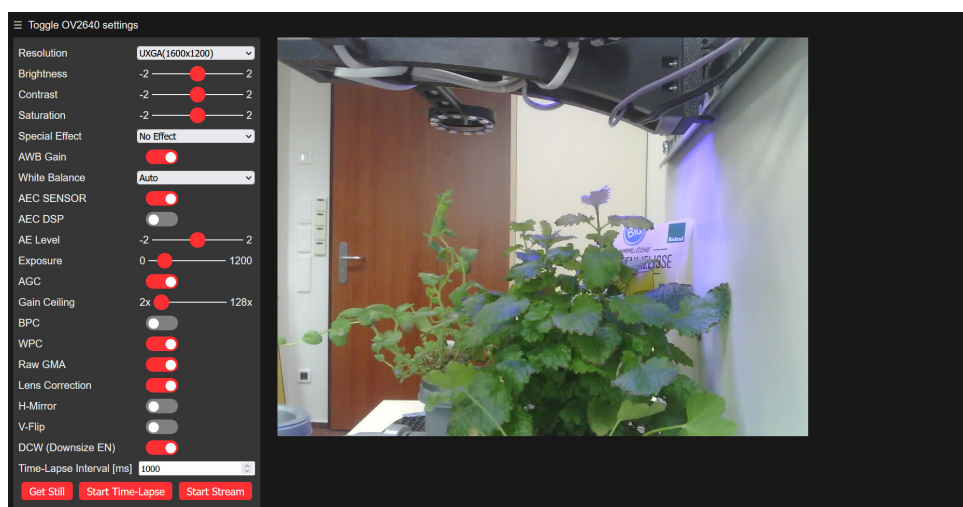
Jelikož se jedná o ESP32 modul, je možné jen nastavit jako *Access point*, nicméně v naší konfiguraci se nám bude hodit modul v režimu *Station*. Po zadání správných přihlašovacích údajů pro přihlášení na místní WiFi síť, se modul připojí na internet a je možné jej spravovat z webového rozhraní na adrese <http://147.32.223.210/>, která odpovídá jeho přiřazené veřejné IP adrese.

Ve webovém rozhraní, které vychází ze základního demo projektu pro tento modul, jsou v levé části zobrazeny ovládací prvky pro nastavení snímání kamery jako například: rozlišení, jas, kontrast a další. Zajímavými vlastnostmi je taky již vestavěná možnost přidání efektu speciálního filtru například:



Obrázek 5.1: Propojení kamerového modulu (vlevo) s *USB-UART* převodníkem (vpravo)

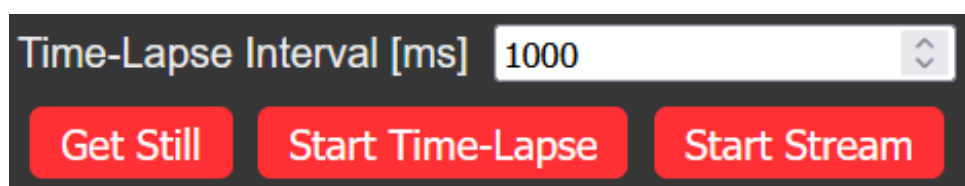
negativ, černobílý filtr, nebo červené, zelené, či modré zabarvení. Uprostřed obrazovky je ponecháno místo pro zobrazování snímků z kamery (obrázek 5.2).



Obrázek 5.2: Webové rozhraní pro ovládání kamerového modulu

5.2 Použití

Hlavní možnosti použití kamery jsou nyní 3: zobrazení jednoho snímku, nastavení a spuštění zaznamenávání časosběru a streamování obrazu z kamery na webový prohlížeč. Všechny 3 lze ovládat tlačítky ve webovém rozhraní (obrázek 5.3).



Obrázek 5.3: Ovládací tlačítka pro jednotlivé funkce kamerového modulu

■ 5.2.1 Jeden snímek

První a nejjednodušší funkcionalitou kamery je pořízení jednoho snímku a zobrazení ho do webového prohlížeče. Pokaždé, kdy klikneme na tlačítko *Get Still* (pořídí nehybný obraz), kamera udělá snímek toho, co právě vidí a přepíše aktuálně zobrazovaný snímek na webu (pokud tam nějaký je).

■ 5.2.2 Časoběry růstu rostlin

Další možností využití kamery je uchovávání snímků pro následné zobrazení dlouhodobého záznamu jako časoběru růstu rostlin. Takovýto záznam je spustitelný opět přímo z webového rozhraní a to tlačítkem *Start Time-Lapse*. Ne vždy ale chceme stejný interval mezi zaznamenáváním jednotlivých snímků. Například pro pozorování pohybu lidí je dobré mít časoběr s intervalem 1 nebo 2 sekundy. Ve chvíli, kdy ale chceme pozorovat rostliny jak se hýbají, je takovýto interval příliš krátký. Dle mého testování je rozumné pořizovat snímky každých několik minut, tedy od 1 až do 10 minut.

Snímky se ukládají na SD kartu, která je součástí modulu ESP32-CAM. Ta se dá poté vyjmout a přenést do počítače, kde pomocí jednoho *ffmpeg.exe* [20] souboru a jednoho příkazu do příkazového řádku vytvoříme ze snímků plynulý film. Jak plynulý je možné nastavit v rámci zadávání příkazu, kdy se může nastavit 1 až 60 snímků za vteřinu videa časoběru. Člověk určí video o 30 snímcích za vteřinu jako dostatečně plynulé, záleží pak, jestli chceme mít video dvojnásobně kratší při použití 60 snímků, namísto právě zmíněných 30. Soubor *ffmpeg.exe*, společně s kódem pro kamerový modul, který jsem následně upravoval pro naše potřeby, jsem získal ze stránek projektu časoběru: <https://github.com/bitluni/ESP32CamTimeLapse>.

■ 5.2.3 Streamování

Když už víme, že kamera umí posílat na web jednotlivé obrázky, proč tuto funkcionalitu neopakovat plynule po sobě a nenechat si vykreslovat živý přenos. Přesně toto splňuje třetí a aktuálně poslední funkcionalita kamerového modulu. Stream se pouští opět pomocí tlačítka, tentokrát s označením *Start Stream*.

Jelikož se jedná o výpočetně velmi náročnou operaci, nemůžeme očekávat plynulý obraz o 30 a více snímcích za vteřinu. Velký vliv na rychlost má také nastavení rozlišení kamery. Nejnižší rozlišení 160x120 se nejvíce blíží plynulému obrazu, zato je tento obraz ale velmi malý a těžko je z něho rozeznat, na co se díváme. Oproti tomu při navoleném rozlišení 1600x1200

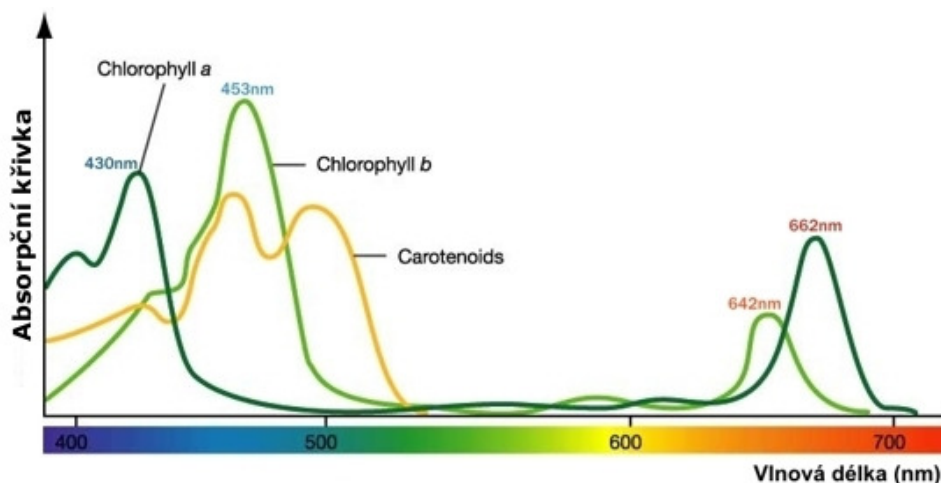
je občas obtížné říci, zda stream vůbec běží. V neposlední řadě má na celou funkci kamerového modulu vliv rychlost připojení k internetu samotného modulu, ale také rychlost připojení prohlížeče, z kterého se snažíme kameru ovládat.

Kapitola 6

Umělé osvětlení pro lepší růst

Při klasickém pěstování rostlin na denním světle si rostliny musí zvyknout na zavedenou dobu svícení a některé mohou být nedostatečně prosvětlené. Proto se do většiny hydroponických systémů přidává umělé osvětlení, které může svítit přesně tak dlouho, jak potřebujeme.

Dalším důležitým světelným aspektem pro pěstování je vysílané spektrum. Pro růst většiny rostlin jsou nejvíce potřeba chlorofyl A, B a karotenoidy. Na grafu v obrázku 6.1 je vidět, jak která barviva vstřebávají světlo. Můžeme si všimnout, že zelené světlo rostliny nevstřebávají téměř vůbec, což je také důvod, proč jsou jejich listy a stonky zelené. Z tohoto grafu můžeme odvodit, že pro růst rostlin je nejlepší modré a červené spektrum. Přesněji modrobílé světlo je nejúčinnější během růstového období rostlin, zatímco červené světlo je blahodárné pro rozvoj květů. Z tohoto důvodu používáme v naší implementaci takovou variantu spektra, která se právě rostlině hodí. Jako ideální nastavení se většinou volí ultrafialové, nebo fialové osvětlení. [10] [21]



Obrázek 6.1: Absorbční křivka rostlin

6.1 Přidání Light Emiting Diod (LED)

Použití LEDek jako formy osvětlení rostlin v projektu hydroponické farmy na naší fakultě jsem zvolil z důvodu velice nízké pořizovací ceny, kvůli dobrému vzhledu při správném zakomponování a také možnosti absolutní nastavitelnosti většiny parametrů. Při zvolení RGB diod, tedy takových, u kterých je možné nastavovat jednotlivé barevné složky zvlášť, nám otevírá ještě mnohonásobně více možností, jak s osvětlením naložit.

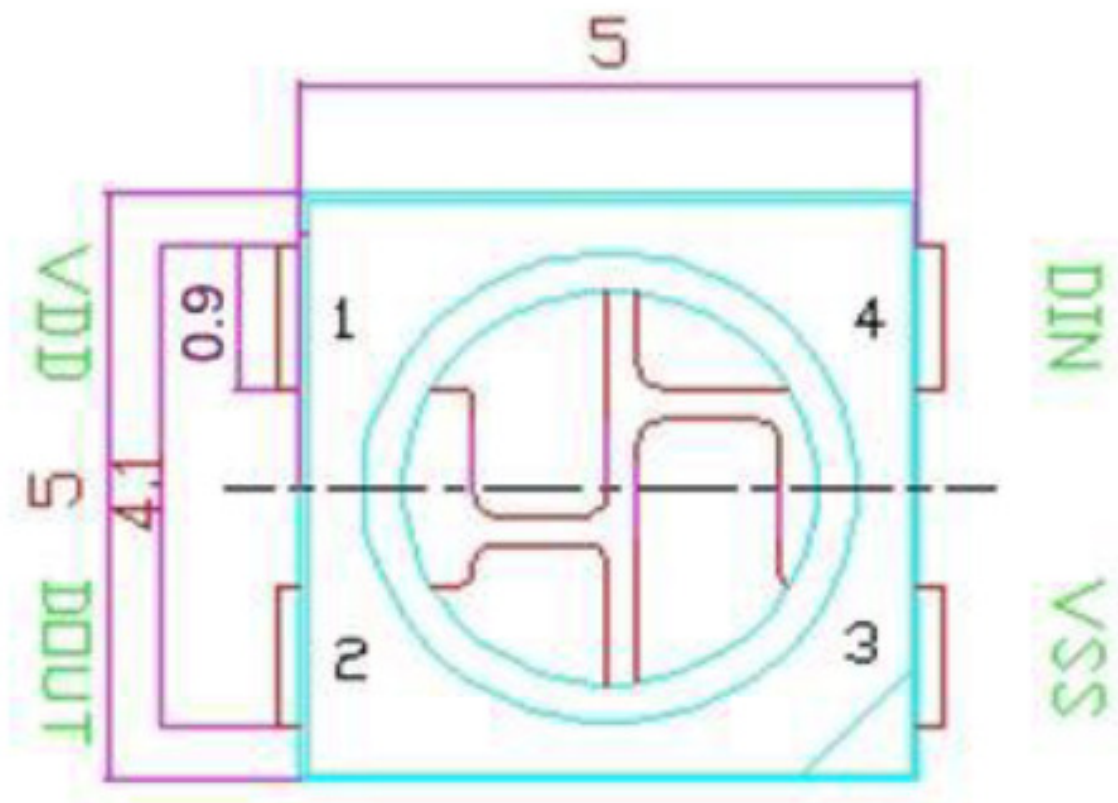
6.1.1 RGB diody WS2812B

Jelikož je náročné používat RGB LEDky v zapojení každé zvlášť, rozhodl jsem se využít již sestavené moduly s 12 NeoPixel WS2812B diodami (viz obrázek 6.2). WS2812B je inteligentní zdroj světla, který se ve skutečnosti skládá z řídicího čipu a RGB čipu zasazeného do 5050 (rozměry 5.0 mm x 5.0 mm) komponentu. Díky vnitřnímu řídicímu čipu je možné řídit takovéto diody za sebe na komponenty typu NeoPixel a stále je řídit pouze jedním PWM (pásmově šířková modulace) výstupem z Arduina. Zároveň lze řídit za sebe i celé NeoPixel komponenty.

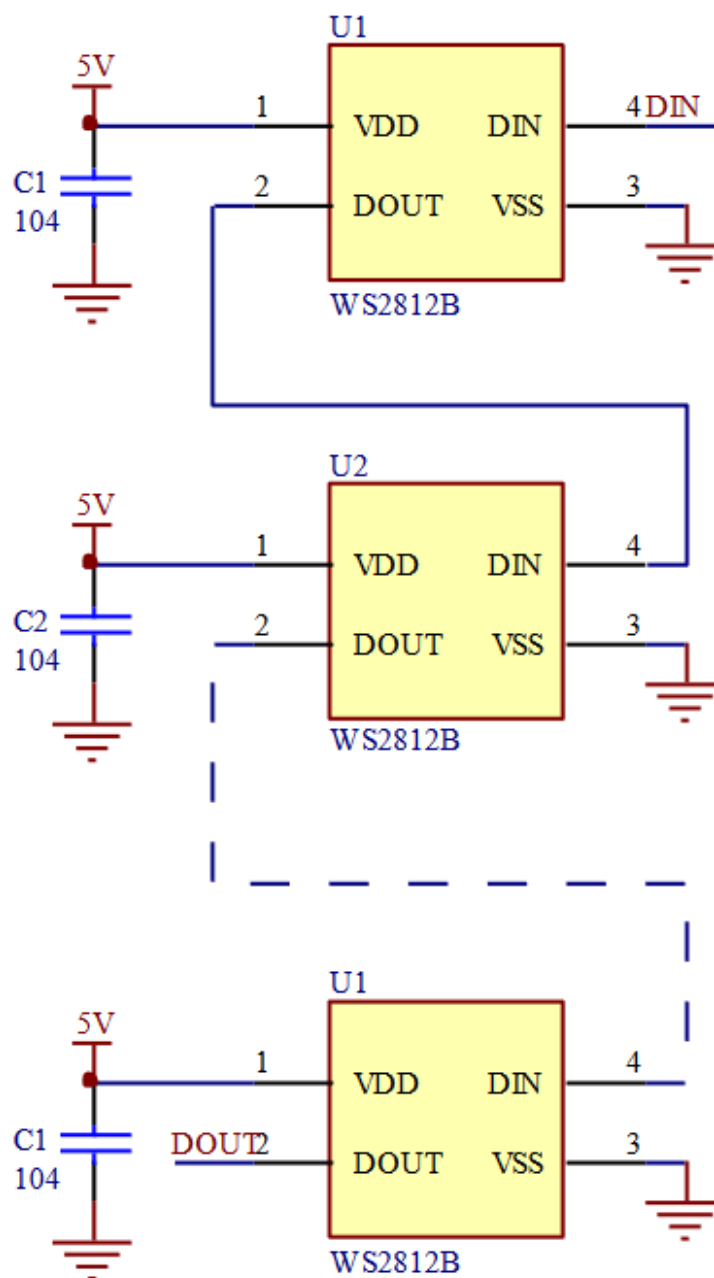


Obrázek 6.2: Kruhový modul NeoPixel s 12 WS2812B diodami zapojenými do série

Na obrázku 6.3 je vidět schéma jedné WS2812B RGB diody a její vstupní a výstupní piny. Na pin VDD se připojí napětí + 5 V, zem se přivede na pin VSS , zatímco řídicí signál přivedeme na pin označený jako DIN . Z pinu $DOUT$ můžeme pak vést řídicí signál na další WS2812B diody, jak je tomu na obrázku 6.4. Stejným stylem jsou zapojeny diody na modulech NeoPixel a pokud si představíme jeden modul jako jednu diodu, jsou stejným stylem zapojeny také celé moduly mezi sebou.



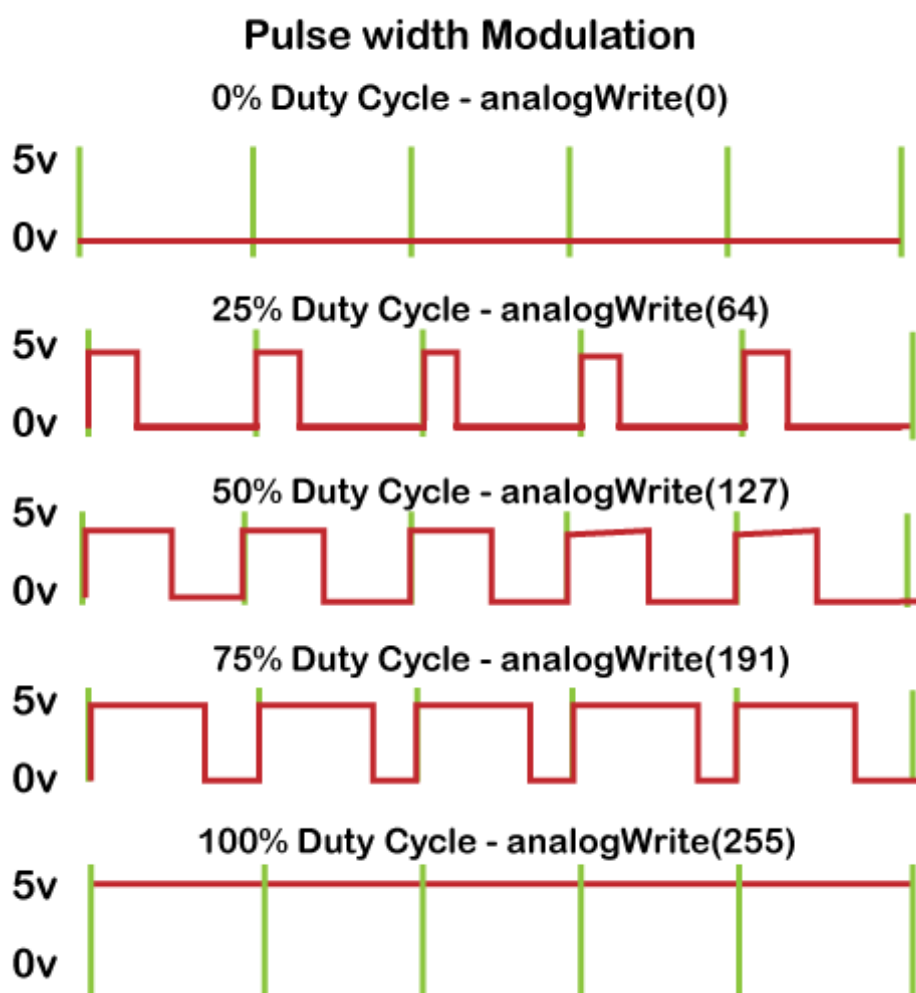
Obrázek 6.3: Schéma znázorňující konfiguraci pinů pro zapojení WS2812B diody



Obrázek 6.4: Schéma znázorňující sériové zapojení WS2812B diod pro řízení po jednom PWM pinu

6.1.2 Řízení osvětlení

Jak už bylo několikrát zmíněno k řízení těchto WS2812B LED se používá PWM. *Pulse Width Modulation (PWM)*, neboli pulzně šířková modulace, je technika pro přenos analogového signálu pomocí dvoustavového digitálního přenosu. Takovýto střídající se signál může simulovat hodnotu napětí mezi plným rozsahem napětí řídicí jednotky (+5 V) a zemí (0 V) pomocí změny délky doby strávené v sepnutém stavu. Této době se říká šířka pásma, neboli *pulse width*. Poměrem šířky pásma ku délce periody získáme střídu signálu. Pokud se tyto pulzy opakují dostatečně rychle, měřením bychom zjistili, že vytváříme analogovou hodnotu v daném rozsahu. Zvolením střídy 50 % docílíme analogové napěťové úrovně odpovídající polovině plného rozsahu. Toto také můžeme vidět na obrázku 6.5, kde je použita Arduino funkce `analogWrite()` spouštěcí PWM výstup.



Obrázek 6.5: Ukázky průběhu napětí na PWM pinu

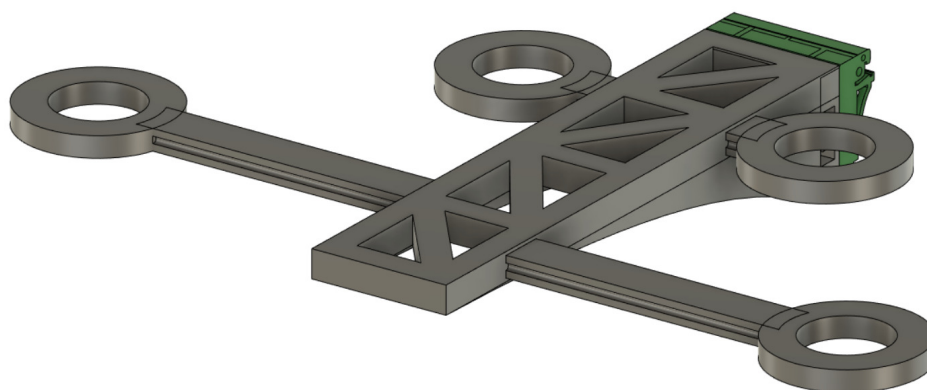
Použitím modulů NeoPixel jsem mohl pro řízení LED osvětlení použít

Hlavní menu	200	Vypnout, zapnout, default	205			Umožňuje úplné vypnutí, zapnutí do předchozího nastavení, nebo nastavit do základu
		Jas	210			Nastavení hodnoty jasu pro všechny diody
		Menu barev	220	Základní barvy	221	Umožňuje zvolit jednu ze základních barev na osvětlení
				Červená	222	Nastavení hodnoty červené složky světla
				Zelená	223	Nastavení hodnoty zelené složky světla
				Modrá	224	Nastavení hodnoty modré složky světla

Tabulka 6.1: Stavy stavového automatu týkající se ovládání LED osvětlení

6.2 3D tištěné osvětlení

V mojí konfiguraci osvětlení jsou použity 4 moduly NeoPixel, každý pro jednu rostlinu. Rostlin se dá sice v projektu farmy pěstovat až 5, nicméně z hlediska estetičnosti a čistě demonstračního účelu jsem ustoupil k tvorbě osvětlení pouze pro 4. Samotné kroužky diod by těžko držely nad rostlinami a kdybych se snažil o nějaké provizorní umístění, nevypadalo by to zcela určitě hezky. Jelikož má projekt farmy sloužit především k prezentačním účelům, rozhodl jsem se pro elegantnější řešení pomocí 3D tisku. Celá konstrukce (viz obrázek 6.6) se skládá z několika dílů, které musely být vytisknuty samostatně a následně propojeny.

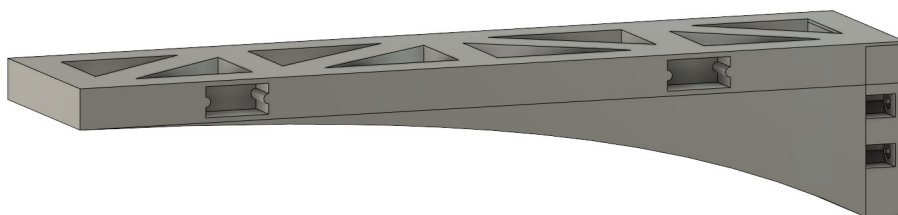


Obrázek 6.6: 3D model držáku LED osvětlení

6.2.1 Hlavní rám

Jak už název sekce napovídá, jedná se o hlavní část konstrukce LED osvětlení, ke které se následně přidělávají ostatní části. Vzhledově a konstrukčně jsem se inspiroval u jiného rámu implementovaného v projektu a vytisknutého na 3D tiskárně.

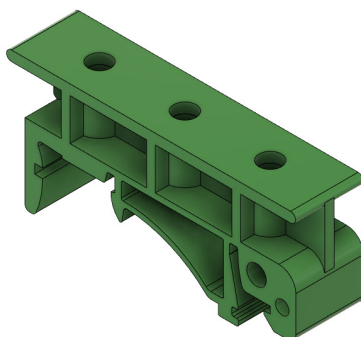
Rozměrově si jsou rámy podobné: 210 x 50 x 42 mm. Tyto rozměry a konstrukční prvky typu trojúhelníkových otvorů a obloukových podpěr zajišťují pevnost a přitom elegantnost celého rámu. Rám disponuje 4 obdélníkovými otvory pro zasazení ramen držících kroužky s diodami a 4 otvory pro vložení M3 matek, kterými se připojuje držák na DIN lištu. Na obrázku 6.7 je zobrazen pouze hlavní rám s důrazem na otvory pro další díly.



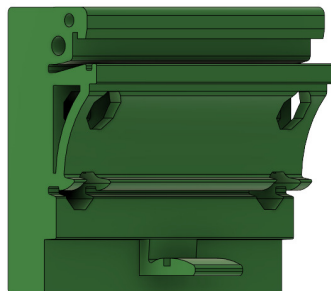
Obrázek 6.7: 3D model hlavního rámu s otvory pro vložení dalších dílů 3D tisku

■ 6.2.2 Držák na DIN lištu

Aby bylo možné hlavní rám nějak přichytit ke konstrukci farmy, rozhodl jsem se pro odjímatelný držák na DIN lištu. Model držáku (obrázek 6.8) jsem stáhl z internetu [22], ale bylo třeba ho velmi upravit, neboť jeho šířka 10 mm byla krajně nevyhovující. Držák jsem tedy roztáhl na potřebnou šířku hlavního rámu, tedy na 50 mm, a upravil jeho výstupky, aby byly dostatečně pevné pro držení konstrukce na DIN liště, ale zároveň aby bylo možné celý model z DIN lišty sundat (upravený držák na obrázku 6.9).

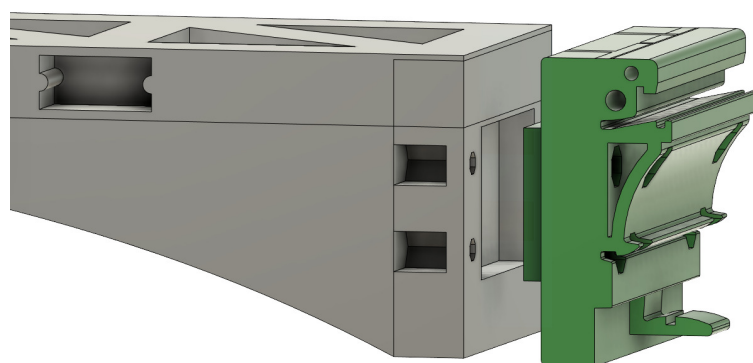


Obrázek 6.8: 3D model držáku na DIN lištu stažený z internetu



Obrázek 6.9: 3D model držáku na DIN lištu po mých úpravách

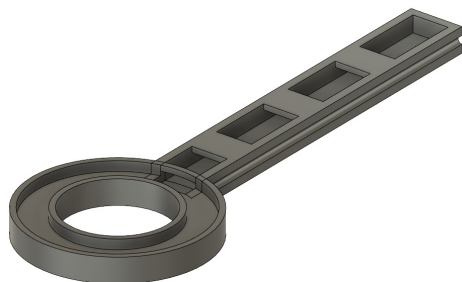
Věděl jsem, že chci tyto dva kusy (rám a držák) spojit pomocí M3 šroubků a matic, proto bylo důležité ponechat na ně otvory, ale znovu jsem se nechal inspirovat konstrukcí obalu pro WiFi modul, který je taktéž vytištěný na 3D tiskárně a skládá se ze dvou dílů. Doplnil jsem tedy z držáku vystupující kvádr, který se zasune do otvoru na zádech hlavního rámu a zajistí, aby se tyto dva díly navzájem nijak nehýbaly. Jak do sebe zapadá držák a hlavní rám, je vidět na obrázku 6.10.



Obrázek 6.10: Model hlavního rámu a držáku s detailem na kvádrový výstupek pro stabilitu

■ 6.2.3 Ramena na LED kroužky

Poslední části potřebné pro konstrukci LED osvětlení naší hydroponické farmy jsou ramena pro držení samotných NeoPixel modulů. Model každého ramena je ve skutečnosti složen ze dvou dílů, které se ale fyzicky vytisknou jako jeden celistvý. Základem je tedy kroužek, do kterého se vsadí NeoPixel modul, a k němu je připojen vodící kanál (rameno), v kterém se prostrčí kabel pro napájení a ovládání LED kroužku. Záběr zespodu na jedno rameno je zobrazen na obrázku 6.11.

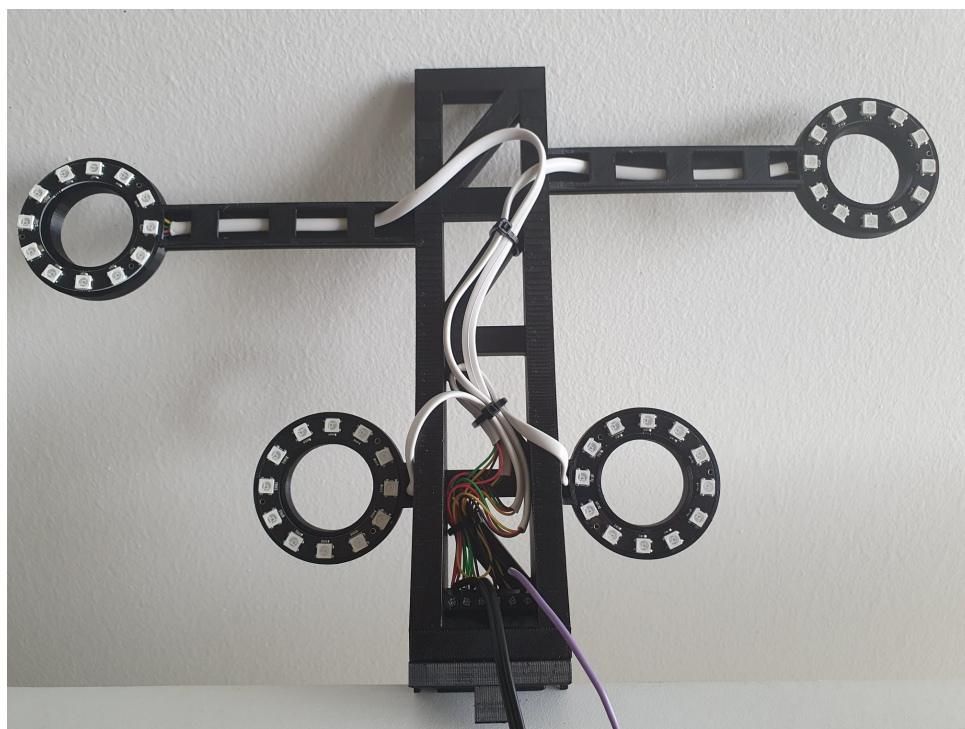


Obrázek 6.11: Model ramene pro uchycení LED modulu NeoPixel

Z obrázku 6.6 je patrné, že jsou dva druhy těchto držáků, jeden s delším a druhý s kratším ramenem. Zároveň si můžeme všimnout, že přední (delší) ramena jsou vůči sobě částečně vyosena. Tyto konstrukční vlastnosti jsou implementovány z důvodu, že takto jsou rozmístěny košíčky, ve kterých rostou rostliny. Takto je vždy jeden LED modul umístěn přímo nad jednou květinou.

■ 6.2.4 Propojení a oživení

Po vytisknutí jednotlivých dílů nebylo příliš složité dát je dohromady. Pro příště by však bylo lepší zavést drobné mezery například do otvorů pro zasunutí jednotlivých ramen, aby do sebe díly zapadly pohodlněji. Takto jsem musel konec každého ramene částečně zúžit brusným papírem. Následně stačilo ke každému NeoPixel modulu připájet část telefonní dvojlinky jako propojovací kabel z konce jednotlivých ramen do středu hlavního rámu (viz obrázek 6.12), kde jsem napájecí vodiče všech modulů zapojil do jedné napájecí svorky a jedné zemnicí svorky. Od svorek zasazených v hlavním rámu je už vedena pouze jedna napájecí dvojlinka. Následně jsem do série zapojil všechny 4 LED kroužky a vstup prvního z nich přivedl na PWM pin řídicí jednotky Arduino MEGA. Výsledné osvětlení je vidět na obrázku 6.13.



Obrázek 6.12: Přivedení kabelů od 4 NeoPixel modulů z ramen do středu hlavního rámu



Obrázek 6.13: Výsledné použití 3D vytisknutého LED osvětlení nad hydroponickou farmou

Kapitola 7

Závěr

Cílem této bakalářské práce bylo vytvořit stručný přehled systémů pro hydroponické pěstování v domácím prostředí a práce na již existujícím prototypu. V druhé kapitole byly popsány typy hydroponických farem, čím se vyznačují a jak zařizují přínos závlahy a živin rostlinám. Většina takovýchto farem řeší tyto problémy použitím živného roztoku, který má lepší vlastnosti než obyčejná voda a který je povětšinou kontrolován přidruženou automatizací.

Hlavní částí této práce bylo rozšíření prototypu hydroponického systému na Fakultě elektrotechnické. Konkrétně o propojení komunikačním kanálem RS485 hlavní řídicí desku s WiFi modulem ESP32. Data přenášená po komunikační lince jsou zobrazována na webovém serveru běžícím na WiFi modulu. Server je koncipován jako asynchronní a stará se o přicházející požadavky z prohlížeče na pozadí. Z webového rozhraní je možné posílat také informace zpátky do řídicí jednotky jako například spuštění procesu zalévání.

Ze serveru běžícího na ESP32 WiFi modulu je možné odkazem přejít na webové rozhraní kamerového modulu ESP32-CAM. Tento modul je schopný se připojit k WiFi a do prohlížeče připojeného na jeho IP adresu streamovat živý přenos z hydroponické farmy. Za druhé může kamera ukládat snímky na microSD kartu a z nich můžeme v počítači vytvořit video růstu rostlin. Obě webová rozhraní by bylo v budoucnu dobré spojit v rámci jiného serveru, na který by se dala ve větším množství ukládat data z měření na senzorech, či právě zmíněné časosběry rostlin.

Hmatatelným výstupem mé práce je konstrukce vytisknutá na 3D tiskárně a zapojení RGB LED osvětlení pomocí modulů NeoPixel. Toto plně nastavitelné osvětlení při správné konfiguraci zvyšuje účinnost příjmu světla rostlinami a výrazně tak podporuje jejich růst. Konstrukce, kterou jsem sám navrhl, se skládá z několika dílů, které se spojují pomocí zasunutí do sebe nebo spojením M3 šroubky. Takovéto členění nám umožňuje jednotlivé části vyměnit, pokud dojde k jejich poškození například ulomením, což se u 3D tisků občas stává.

Projekt hydroponické farmy je nyní už pohodlně ovladatelný prostřednictvím webového serveru, ale do budoucna bychom mohli docílit plné automatizace udržování živného roztoku. Tomu bychom pomohli například přidáním senzorů pro měření teploty, či vodivosti roztoku a na základě jejich hodnot pak následnou implementací automatického dávkování živin nebo přitápění.



Literatura

- [1] KVAPIL, Marek. Potravinová soběstačnost na 465 m². In: *Potravinové zahrady* [online]. [cit. 2022-01-08]. Dostupné z: <https://www.potravinovezahrady.cz/potravinova-sobestacnost-na-465-m2/>
- [2] AL-KODMANY, Kheir. The Vertical Farm: A Review of Developments and Implications for the Vertical City. In: *Buildings* [online]. Chicago, 2018, , s. 37 [cit. 2022-01-08]. ISSN 2075-5309. Dostupné z: <http://www.mdpi.com/2075-5309/8/2/24>
- [3] ALBERY, W.John, Barry G.D. HAGGETT a L.Robert SVANBERG. The development of sensors for hydroponics. In: *Biosensors* [online]. Londýn, 1985, [cit. 2022-01-08]. ISSN 0265928X. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/0265928X85800067>
- [4] Ebb and flow. *Cambridge Dictionary* [online]. Cambridge [cit. 2022-01-09]. Dostupné z: <https://dictionary.cambridge.org/dictionary/english/ebb-and-flow>
- [5] Hydroponie snadno: EBB & FLOW. *Higarden* [online]. Praha [cit. 2022-01-09]. Dostupné z: <https://www.higarden.cz/blog/hydroponie-snadno-ebb-flow/>
- [6] Aeroponie. *Higarden* [online]. Praha [cit. 2022-01-09]. Dostupné z: <https://www.higarden.cz/slovník-pojmu/aeroponie/>
- [7] Aquaponie je budoucnost pěstování potravin. *Future farming* [online]. Brno [cit. 2022-01-09]. Dostupné z: <https://www.futurefarming.cz/aquaponie-je-budoucnost-pestovani-potravin/>
- [8] Arduino Mega 2560 Rev3. *Arduino.cc* [online]. [cit. 2022-01-09]. Dostupné z: <https://store.arduino.cc/products/arduino-mega-2560-rev3>
- [9] Getting Started with the ESP32 Development Board. *Random nerd tutorials* [online]. [cit. 2022-01-09]. Dostupné z: <https://randomnerdtutorials.com/getting-started-with-esp32/>

- [10] KERNER, Matěj. *Automatický systém hydroponického pěstování rostlin*. Praha, 2021. Bakalářská práce. České vysoké učení technické. Vedoucí práce Doc. Ing. Stanislav Vítek, Ph.D.
- [11] ME NO DEV. ESPAsyncWebServer. *GitHub* [online]. 2019 [cit. 2022-05-18]. Dostupné z: <https://github.com/me-no-dev/ESPAsyncWebServer#espasyncwebserver>
- [12] SPIFFS Filesystem. *Espressif* [online]. Shanghai, 2022 [cit. 2022-05-19]. Dostupné z: <https://docs.espressif.com/projects/espressif/en/latest/esp32/api-reference/storage/spiffs.html>
- [13] SPIFFS in ESP32. *Tutorialspoint: simply easy learning* [online]. 2022 [cit. 2022-05-19]. Dostupné z: https://www.tutorialspoint.com/esp32_for_iot/esp32_for_iot_spiffs_storage.htm
- [14] HTML Introduction. *W3Schools* [online]. 2022 [cit. 2022-05-18]. Dostupné z: https://www.w3schools.com/html/html_intro.asp
- [15] CSS Introduction. *W3Schools* [online]. 2022 [cit. 2022-05-18]. Dostupné z: https://www.w3schools.com/css/css_intro.asp
- [16] JavaScript Introduction. *W3Schools* [online]. 2022 [cit. 2022-05-18]. Dostupné z: https://www.w3schools.com/js/js_intro.asp
- [17] POUCHA, Pavel. Komunikace pro průmyslových linkách RS485 a RS422. *PaPouch* [online]. 2022 [cit. 2022-05-19]. Dostupné z: <https://papouch.com/komunikace-pro-prumyslovych-linkach-rs485-a-rs422-p3735/>
- [18] HTML Forms. *W3Schools* [online]. 2022 [cit. 2022-05-18]. Dostupné z: https://www.w3schools.com/html/html_forms.asp
- [19] Vývojová deska ESP32-CAM. *Drátek.cz* [online]. Havlíčkův Brod [cit. 2022-05-19]. Dostupné z: https://navody.dratek.cz/navody-k-produktum/vyvojova-deska-esp32-cam.html?gclid=Cj0KCQjw1ZeUBhDyARIsAOzAqQJlrTan0ACb3kuMJXJIAHE-Bo0cpnq4DLsmzrOPHUhfuQVU3AWSNwaAmp9EALw_wcB
- [20] BITLUNI. ESP32CamTimeLapse. *GitHub* [online]. 2020 [cit. 2022-05-19]. Dostupné z: <https://github.com/bitluni/ESP32CamTimeLapse>
- [21] Osvětlení rostlin. *Growshop* [online]. KOLB.CZ, 2022 [cit. 2022-05-19]. Dostupné z: <https://www.growshop.cz/katalog/seznam/osvetleni-rostlin>
- [22] ASHER, Joey. DRG-10 35mm DIN Rail Mount. *GrabCAD* [online]. 2021 [cit. 2022-05-19]. Dostupné z: <https://grabcad.com/library/drg-10-35mm-din-rail-mount-1>

CITACE OBRÁZKŮ:

- [23] ADMIN. NFT system. In: *Marijánka.cz* [online]. [cit. 2022-01-09]. Dostupné z: <https://marijanka.cz/wp-content/uploads/2015/08/hdropo5.jpg>
- [24] JAY. Aeroponie. In: *Roots: časopis s kořeny* [online]. 2018 [cit. 2022-01-09]. Dostupné z: <https://casopisroots.cz/wp-content/uploads/2018/06/Bez-názvu.png>
- [25] Typical operating circuit. In: MAX481/MAX483/MAX485/MAX487–MAX491/MAX1487 [online]. maxim integrated, 2014 [cit. 2022-05-20]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/MAX1487-MAX491.pdf>
- [26] MISCHINATI, Renzo. Esp32-cam upload sketch connection scheme 5v. In: *Renzo Mischianti: A blog of digital electronics and programming* [online]. Hueman Pro, 2022 [cit. 2022-05-19]. Dostupné z: <https://www.mischianti.org/wp-content/uploads/2021/03/esp32-cam-upload-sketch-connection-schema-5v.jpg>
- [27] POSPÍŠIL, Vojtěch. Absorpční křivka. In: *Pěstování, rady a zajímavosti o* [online]. Webnode [cit. 2022-01-09]. Dostupné z: <https://d6scj24zvfbo.cloudfront.net/78d5f52265c4cc74adc403a11c5a168c/200000088-adbcbabeb6c/450/absor1.jpg?ph=82792f139e>
- [28] RGB LED kruh 12 x NeoPixel WS2812B. In: *Drátek.cz* [online]. Havlíčkův Brod [cit. 2022-05-19]. Dostupné z: <https://dratek.cz/photos/produkty/d/7/7695.jpg?m=1563964721>
- [29] PIN configuration. In: WS2812B: Intelligent control LED integrated light source [online]. Worldsemi [cit. 2022-05-19]. Dostupné z: <https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf>
- [30] Typical application circuit. In: WS2812B: Intelligent control LED integrated light source [online]. Worldsemi [cit. 2022-05-19]. Dostupné z: <https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf>
- [31] Pulse Width Modulation. In: *Arduino Forum: Dim LED on 74HC595 by setting OE-Pin via PWM?* [online]. 2021 [cit. 2022-05-20]. Dostupné z: <https://aws1.discourse-cdn.com/arduino/original/4X/e/6/d/e6dcde9724b0bf41c99e967353e11b7242161d9b.png>