



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

## ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

## NÁVRH A REALIZACE ŘÍDÍCÍ JEDNOTKY VNITŘNÍHO PROSTŘEDÍ AKVÁRIA

DESIGN AND IMPLEMENTATION OF A CONTROL UNIT FOR THE AQUARIUM ENVIRONMENT

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Lukáš Petrák

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Krejčí,  
Ph.D.

BRNO 2022

# Zadání bakalářské práce

Ústav: Ústav mechaniky těles, mechatroniky a biomechaniky  
Student: **Lukáš Petrák**  
Studijní program: Aplikované vědy v inženýrství  
Studijní obor: Mechatronika  
Vedoucí práce: **Ing. Petr Krejčí, Ph.D.**  
Akademický rok: 2021/22

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

## Návrh a realizace řídicí jednotky vnitřního prostředí akvária

### Stručná charakteristika problematiky úkolu:

Bakalářská práce je zaměřena na návrh a realizaci řídicí jednotky pro automatické řízení provozu akvária. Řídicí jednotka bude schopna dle nastavených kritérií monitorovat a regulovat základní parametry prostředí akvária (teplota, osvětlení atd.). Sledované parametry budou zobrazeny na LCD řídicí jednotky. Řídicí jednotka bude dále poskytovat uživateli možnost přístupu k měřeným veličinám prostřednictvím internetové sítě s možností modifikace parametrů.

### Cíle bakalářské práce:

1. Provedení průzkumu trhu v oblasti automatického řízení vnitřního prostředí akvárií.
2. Návrh řídicí jednotky umožňující řízení teploty a osvětlení akvária.
3. Realizace funkčního vzorku řídicí jednotky.

### Seznam doporučené literatury:

Handbook of Modern Sensors: Physics, Designs, and Applications 5th ed. 2016 Edition, Springer International Publishing, Switzerland 2016

ZÁHLAVA, V.: Návrh a konstrukce plošných spojů, Vydavatelství BEN, 2011.

VALÁŠEK, M. a kol.: Mechatronika, Vydavatelství ČVUT Praha, 1995.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2021/22

V Brně, dne

L. S.

---

prof. Ing. Jindřich Petruška, CSc.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

## **Abstrakt**

Tato bakalářská práce se zabývá návrhem a realizací prototypu řídicí jednotky pro ovládání vnitřního prostředí akvária. Cílem této práce je vytvořit zařízení, které bude ovládat osvětlení akvária pomocí dostupných LED pásků různých barev, dále bude udržovat teplotu vody v akváriu v rozsahu nastaveném uživatelem. Následně bude na SD kartu zaznamenávat průběh osvětlení, teploty vody ze tří čidel a času, kdy byly tyto hodnoty získány. Tyto naměřené veličiny je možné získat a zařízení ovládat pomocí webového rozhraní z běžného prohlížeče. Zařízení by mělo být realizovatelné z běžně dostupných modulů a součástek a svými parametry by mělo uspokojit jak začínajícího, tak i pokročilého akvaristu, který si z analýzy naměřených hodnot stavu vnitřního prostředí akvária nastaví řídicí jednotku podle svých potřeb.

## **Klíčová slova**

ESP 32, akvárium, automatizace, mikrokontroler, internet

## **Abstract**

This bachelor thesis deals with the design and implementation of a prototype control unit for controlling the internal environment of an aquarium. The aim of this work is to create a device that will control the aquarium lighting using available LED strips of different colors, as well as maintain the aquarium water temperature within the range set by the user. It will then record on an SD card the progress of the lighting, the water temperature from the three sensors and the time when these values were obtained. These measurements can be retrieved and the device can be controlled via a web interface from a standard browser. The device should be realizable from commonly available modules and components and its parameters should satisfy both the beginner and the advanced aquarist who, from the analysis of the measured values of the aquarium's internal environment, will adjust the control unit according to his needs.

## **Keywords**

ESP 32, aquarium, automation, microcontroller, internet

## **Bibliografická citace**

PETRÁK, Lukáš. *Návrh a realizace řídicí jednotky vnitřního prostředí akvária*. Brno, 2022. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/137133>.  
Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Petr Krejčí.

## **Čestné prohlášení**

Prohlašuji, že jsem bakalářskou práci na téma „Návrh a realizace řídicí jednotky vnitřního prostředí akvária“ zpracoval samostatně s použitím informačních zdrojů uvedených v seznamu.

V Brně dne: .....

.....  
Lukáš Petrák

## **Poděkování**

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Petrovi Krejčímu, Ph.D. za ochotu a trpělivost při psaní bakalářské práce. Také velké poděkování je mé rodině, která při mně stála a hlavně tátovi, který mi poskytl mnoho cenných informací.

# Obsah

<b>1</b>	<b>Úvod .....</b>	<b>6</b>
<b>2</b>	<b>Rešerše v oblasti řídicích jednotek akvária .....</b>	<b>7</b>
<b>3</b>	<b>Formulace problému a cíle řešení .....</b>	<b>9</b>
<b>4</b>	<b>Návrh řídicí jednotky pro ovládání teploty a osvětlení (OTO) .....</b>	<b>10</b>
4.1	Výběr platformy a periferních modulů .....	11
4.1.1	Výběr MCU a periférií.....	11
4.1.2	Vybrané moduly a čipy .....	11
4.1.3	Modul ESP32-Wroom-32 .....	12
4.2	Modul SD karty .....	13
4.3	Modul DS3231 + paměť EEPROM AT24C32 .....	14
4.4	Teplotní čidla DS18B20 od firmy Maximintegrated .....	17
4.5	Modul PWM ovládání Led pásků .....	19
4.6	Modul Relé.....	21
4.7	Stavový LCD displej .....	22
4.8	Napájecí zdroj .....	23
4.9	Tlačítko .....	24
<b>5</b>	<b>Programová část řídicího systému .....</b>	<b>25</b>
5.1	Úvodní část programu .....	25
5.2	Počáteční nastavení a inicializace jednotlivých periférií .....	26
5.2.1	Teplotní čidla DS18B20 .....	26
5.2.2	Osvětlení .....	26
5.2.3	SD karta .....	27
5.2.4	Nastavení výstupů pro ovládání topení, UV sterilizéru a vzduchování....	28
5.2.5	Spuštění a konfigurace sítě WIFI.....	28
5.2.6	Nastavení času .....	30
5.2.7	Zobrazované údaje na stavovém displeji .....	30
5.3	Web server .....	32
5.4	Webové rozhraní .....	37
5.4.1	Webová stránka pro nastavení hodnot .....	38
5.4.2	Webová stránka pro ukládání dat.....	40
<b>6</b>	<b>Ověření funkčnosti.....</b>	<b>42</b>
<b>7</b>	<b>Závěr .....</b>	<b>45</b>
<b>8</b>	<b>Seznam použitých zdrojů .....</b>	<b>47</b>
	<b>Seznam použitých zkratk .....</b>	<b>48</b>
	<b>Seznam příloh.....</b>	<b>49</b>



# 1 Úvod

Téměř každý, zejména děti, po zhlednutí krásných barevných akvárií ať už na internetu nebo na různých výstavách zatouží mít něco takového doma. Brzy však po jeho pořízení zjistí, že není vůbec lehké a jednoduché docílit toho, aby i jejich akvária vypadala tak úžasně. Kromě znalostí potřeb jednotlivých druhů rostlin i živočichů je potřeba pro zdárný vývoj života v akváriu i technika, která pomáhá udržovat přijatelné životní prostředí pro jeho obyvatele. Zatím co znalosti a zkušenosti si musí každý získat sám ať už z literatury, od kamaráda akvaristy nebo vlastními chybami, technické vybavení v různých variantách je možno zakoupit ve specializovaných prodejnách. Toto je však poměrně nákladné, a ne vždy splňuje konkrétní požadavky a očekávání. Z tohoto důvodu vznikla tato práce, která si klade za cíl vytvořit návrh poměrně jednoduchého, modulárního a snadno ovladatelného zařízení pro kontrolu a nastavení základních aspektů prostředí v akváriu jako je světlo (jeho intenzita a barva), teplota vody a ukládání těchto parametrů pro následnou analýzu při problémech s prosperitou akvarijního prostředí. Tímto postupem, kromě dodávání patřičných živin, je možné zjistit při jakém druhu světla a teploty se nejlépe daří rostlinám.

V této práci jsou použité dostupné a cenově přijatelné komponenty, které jsou běžně k sehnání, a tak si tuto jednotku může sestavit zručný kutil svépomocí.

Jedná se o zařízení spíše pro experimentální využití, na jehož základě je možné provést jeho úpravy a další rozšíření.

Bakalářská práce je zaměřena na návrh a realizaci řídicí jednotky vnitřního prostředí akvária. Zahrnuje vytvoření www stránek, které jsou vstupní branou pro komunikaci uživatele s řídicí jednotkou OTO. Ta provádí zpracování dat, jejich zobrazení na displej a uložení na SD kartu, měření teploty pomocí teplotních čidel, ovládání třech zařízení pro úpravu vody a načítání dat barevné škály osvětlení z SD karty. Součástí bakalářské práce je i průzkum trhu v oblasti automatického řízení akvarijního prostředí.

## 2 Rešerše v oblasti řídicích jednotek akvária

Na základě stanovených parametrů v kapitole Formulace problému a cíle řešení byla vybrána dostupná zařízení, která by svými parametry z části odpovídala stanoveným požadavkům. Po shlédnutí webových stránek několika prodejců v České republice se nenašlo řešení, které by plně odpovídalo daným kritériím. Z hlediska nároků na osvětlení se mu nejvíce blížila tato zařízení:

Prvním výrobkem je Aquatantis Luminus dvoukanálový kontrolér. Ve firmě Sklorex stojí 2 599 Kč.

Jeho parametry jsou [1]:

- Dva samostatně napájené nezávislé kanály (zatížitelnost každého maximálně 5 A, kontrolér tedy je schopen proudové zátěže celkem 10 A).
- Jeden až čtyři intervaly pro každý kanál pro vytvoření efektů východu, západu slunce a různých denních a nočních fází (simulace měsíčního svitu)
- Nastavitelná světelná intenzita (0 % až 100 %).
- Doba postupného stmívání a rozsvícení nastavitelná od 10 do 90 minut
- Simulace „bouře“ (pomocí aplikace).
- Použitelné pro jakékoliv LED osvětlení 12 nebo 24 V.
- Vnitřní paměť ukládá nastavené programy pro případ výpadku proudu
- Osvětlení můžete programovat i přes mobilní aplikaci (Android i iOS).

Druhý výrobek je Aquatantis EasyLed Control 2 Plus stmívač. Je určen pro regulaci dvou až čtyř světel Aquatantis. Ve firmě Sklorex stojí 1 961 Kč [2].

- Umožňuje postupného rozsvícení a stmívání v určitou hodinu (spínací hodiny se stmívačem).
- Rozsvícení i stmívání je přednastavené na hodinový postupný přechod od nulového do plného výkonu a naopak.
- Oba výstupy lze nastavit individuálně, pro každý lze nastavit jeden nebo dva intervaly.
- Regulace intenzity osvětlení 0-100 %-můžete omezit plný výkon osvětlení.
- Vnitřní baterie-při výpadku proudu je nastavení zachováno.
- Zapojuje se mezi zdroj a svítidla.

Třetím výrobkem je Programovatelný stmívač pro LED osvětlení s WIFI ovládáním. Ve firmě INVITAL Aqua s.r.o. stojí 1 490 Kč [3].

- Programovatelný časový ovladač pro LED osvětlení, s možností nastavení času, intenzity a mixace 5 kanálů nezávisle na sobě v 5 nezávislých sekcích v časové ose.
- Nastavení lze provádět bezdrátově, a to pomocí aplikace v PC nebo mobilu.
- Nastavení 50 programů, nastavení lze provádět po 30 minutách, intenzita po 5 %.

Z těchto výrobků se jeví jako nejlepší z hlediska cena/výkon (funkčnost) třetí výrobek. Nutno však podotknout, že vybrané zařízení slouží pouze k ovládání osvětlení a pro kontrolu teploty vody je nutné je doplnit dalším samostatným zařízením, což celou situaci komplikuje. Z těchto důvodů bylo přistoupeno k realizaci vlastního řešení, které bude splňovat stanovené cíle.

### 3 Formulace problému a cíle řešení

Cílem práce je realizace řídicí jednotky vnitřního prostředí akvária. Na základě průzkumu trhu bylo zjištěno, že řídicím jednotkám chybí větší funkčnost a variabilita možností zapojení, a proto v bakalářské práci bude řešen návrh nového řídicího systému akvária.

Požadavky na zařízení:

- Ovládání osvětlení v závislosti na časovém průběhu dne.
- Měření teploty vody.
- Ovládání tří výstupních zařízení (bylo zvoleno vzduchování, UV sterilizer a teplotní těleso na ohřev vody v akváriu).
- Regulaci teploty vody (pouze ohřev).
- Záznam naměřených hodnot a jejich ukládání na SD kartu.
- Webový server pro zobrazení a uložení hodnot a obsluhu celého zařízení (komunikace probíhá prostřednictvím WIFI a webových stránek).

K dosažení vytyčených cílů je potřeba postupovat následovně:

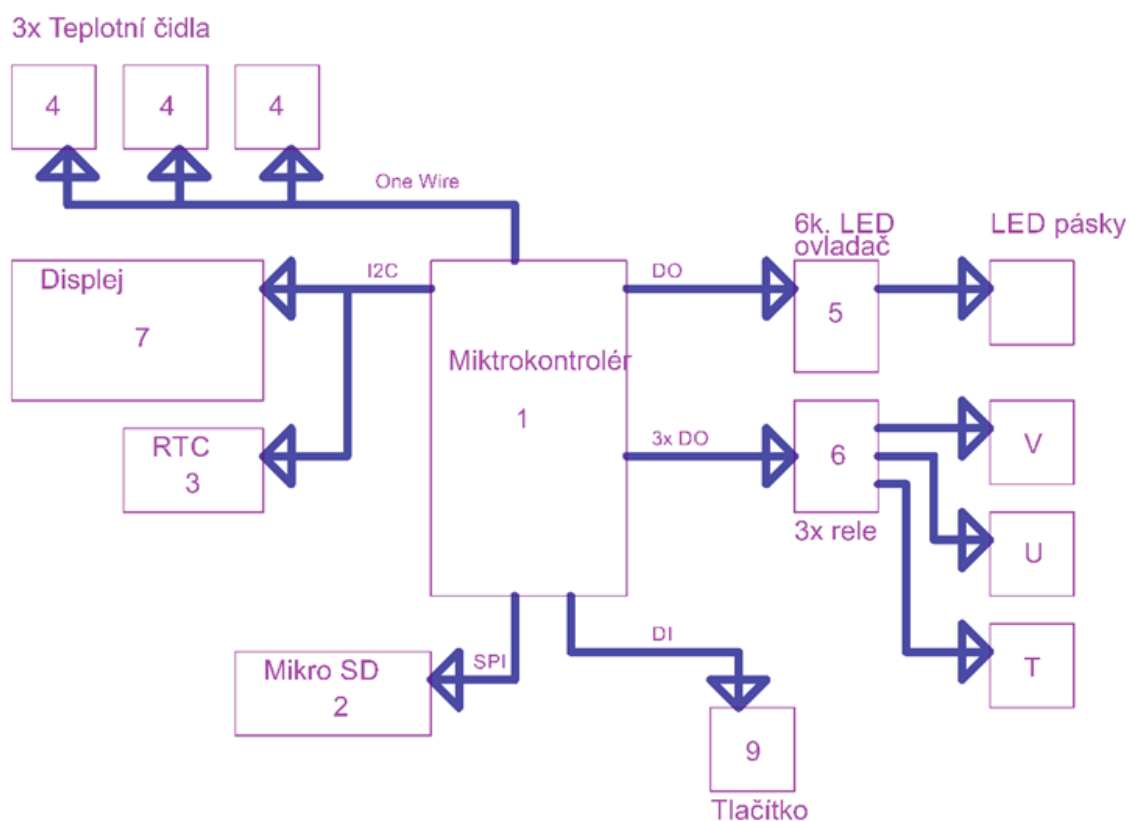
- Navrhnout vhodnou strukturu řídicí jednotky.
- Vybrat mikrokontroler, který bude tuto jednotku ovládat.
- Zvolit vhodné periférie.
- Vytvořit prototyp zařízení na nepájivém kontaktním poli.
- Naprogramovat mikrokontroler a webové rozhraní.
- Otestovat funkčnost řídicí jednotky a odstranit případné chyby.
- Vytvořit finální verzi zařízení na univerzální desce a její odzkoušení.

## 4 Návrh řídicí jednotky pro ovládání teploty a osvětlení (OTO)

Na základě požadavků a vlastností zařízení pro ovládání vnitřního prostředí akvária bylo vytvořeno následující blokové schéma na Obr. 1.

Zařízení pro ovládání akvária se skládá z těchto funkčních celků:

1. mikrokontroler
2. modul SD karty
3. modul reálného času RTC DS3231 + paměť EEPROM AT24C32
4. 3 teplotní čidla DS18S20
5. výkonový modul založený na PWM ovladači LED WS2811
6. modul relé spínajících výstupy.
7. stavový LCD displej
8. napájecí zdroj
9. tlačítko



Obr. 1: Blokové schéma řídicí jednotky

## 4.1 Výběr platformy a periferních modulů

Z důvodu dřívějších kladných zkušeností s programováním jednočipových MCU na platformě Arduino byl výběr zaměřen právě na ni. Tato platforma je podporovaná širokou komunitou uživatelů a vývojářů, a tak pro ni existuje velké množství nejrůznějších periferních zařízení a knihoven. Také výběr podporovaných MCU je dosti široký a zahrnuje jak velice jednoduché 8bitové mikroprocesory, tak i mnohem výkonnější 32bitové. Tato platforma je flexibilní a jednotlivé moduly lze přímo pospojovat mezi sebou jednoduchým zacvaknutím do sebe nebo pomocí propojovacích vodičů (což šetří čas, neboť není nutné nic pájet na univerzální propojovací desku plošných spojů). Také změna zapojení je velice snadná. Jednou z mála nevýhod těchto modulů je jejich kvalita, což je však vyváženo jejich cenou a dostupností.

### 4.1.1 Výběr MCU a periférií

Mezi jeden z nejpoužívanějších MCU patří modul Arduino UNO s mikroprocesorem ATmega328P.

Tento modul nebyl vyhovující z těchto důvodů:

- Malá operační paměť RAM (8KB).
- Nedostatečná velikost flash paměti (32KB), nepojmula by programové vybavení, jehož velikost se vyšplhala až na 105 KB.
- Pro komunikaci přes bezdrátovou WIFI síť je potřeba dalšího externího modulu, který by komplikoval zařízení.

Další možností byl modul s čipem ESP8266, který již obsahuje WIFI modul a jeho výkon je dostatečný na provoz web serveru. Bohužel se později ukázalo, že i 32KB RAM je pro tento projekt nedostatečných. Na základě uvedených skutečností byl vybrán novější čip ESP32 od stejné firmy, který vyhověl nejen rychlostí a množstvím integrovaných funkcí, ale hlavně velikostí paměti RAM (520KB) a paměti flash (4 MB).

### 4.1.2 Vybrané moduly a čipy

- Pro uchování informace o čase nezávisle na ESP32: modul reálného času RTC DS3231.
- Pro měření teploty vody: 3 teplotní čidla DS18B20 od firmy Maximintegrated.
- Pro ovládání LED pásků: modul založený na PWM ovladači Led WS2811.
- Pro ovládání výstupů: modul obsahující 3 relé pro spínání napětí 230 V.
- Pro indikaci stavu zařízení: standardní LCD modul 4×20 znaků se sběrnici I2C založený na čipu HD44780.
- Pro ukládání dat byl použit běžný modul pro mikro SD kartu s SPI rozhraním.
- To celé napájí zdroj určený pro PC, který vyhovuje jak napětově (3.3 V, 5 V, 12 V), tak výkonově (max 350 W).

### 4.1.3 Modul ESP32-Wroom-32

Je srdcem celého zařízení.

- Stará se o časovou posloupnost jednotlivých kroků.
- Zajišťuje komunikaci mezi teplotními čidly.
- Provádí vyhodnocení teploty podle dat z SD karty.
- Zjišťuje čas pomocí NTP serveru.
- Spravuje záložní zdroj času RTC.
- Načítá z SD karty přihlašovací údaje pro WIFI síť.
- Ovládá osvětlení Led pásků podle dat z SD karty v závislosti na čase.
- Provádí zápis údajů na SD kartu pro pozdější využití.
- Obsluhuje WEB server pro zobrazení a nastavení hodnot.
- Na LCD displeji zobrazuje základní údaje.
- Ovládá výstupy pro zapínání a vypínání akvaristických zařízení.

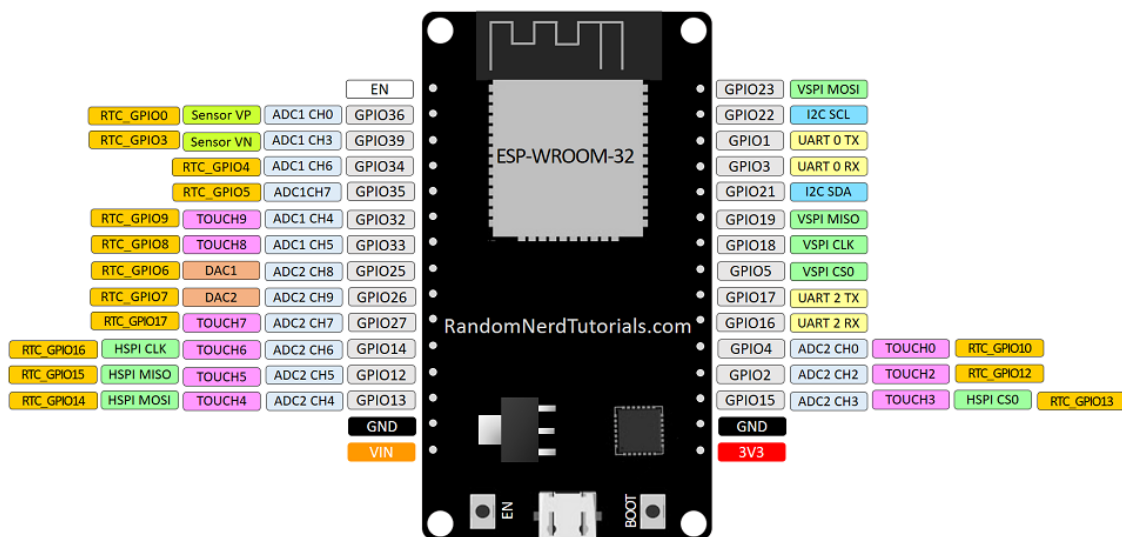
Tento modul obsahuje [4] [5]:

- Dva 32bitové mikroprocesory.
- Interní paměť zahrnující 448 KB ROM pro bootování a základní funkce, 520 KB na čipu SRAM pro data a instrukce a 8 KB SRAM v RTC pro ukládání dat.
- Externí flash má velikost 4 MB.
- 34 GPIO pinů, které mohou vykonávat různé funkce pouhou změnou v nastavení příslušných registrů. Existuje několik typů GPIO (Digitální (pouze), Analogové, Kapacitní a Dotykové). Analogové GPIO a GPIO s kapacitním dotykem lze nastavit na digitální GPIO. Většina digitálních GPIO lze nakonfigurovat jako interní pull-up či pull-down nebo dokonce na vysokou impedanci.
- 3 UART rozhraní a to UART0, UART 1 a UART 2, které umožňují asynchronní komunikaci (RS 232 a RS 485) a podporu IrDA.
- 2 rozhraní sběrnice I2C, které mohou sloužit jako Master nebo jako Slave. Rozhraní I2C podporuje Standardní režim (100 Kbit/s) a Rychlý režim (400Kbit/s).
- 3 SPI (SPI, HSPI and VSPI) v režimu Slave nebo Master. Komunikační režim může být nastaven na duplex anebo polo duplex do maximálního kmitočtu 80 MHz.
- WIFI s přenosovou rychlostí 150 Mbps a výstupním výkonem 20 dBm na anténě. Odpovídá standardu 802.11 b/g/n (2.4 GHz).

Tento modul je určen pro připojení na PCB což pro vývoj není ideální. Proto byl vybrán vývojový modul DevKitV1 na Obr. 2, který již obsahuje převodník USB/seriál pro komunikaci s ladícím programem, stabilizátor pro napájení 3,3 V, tlačítka pro reset a nahrání programu do paměti flash. Jednotlivé vstupy a výstupy jsou připojeny na piny, které lze přímo zapojit do nepájivého kontaktního pole.

## ESP32 DEVKIT V1 – DOIT

version with 30 GPIOs

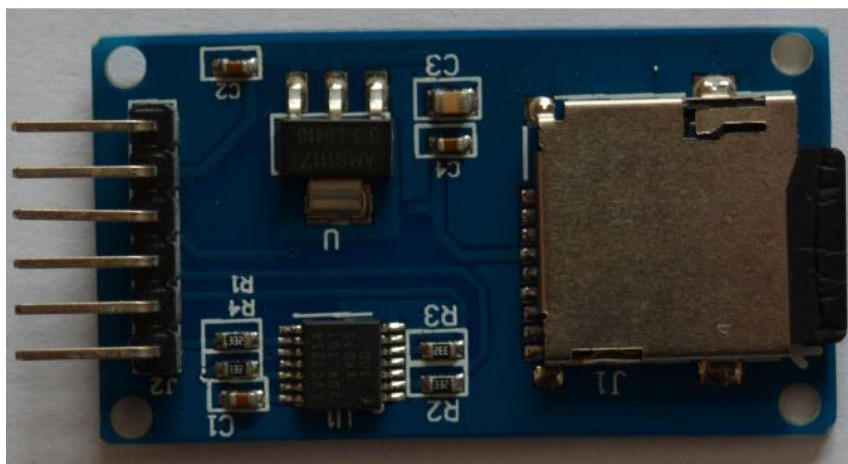


Obr. 2: ESP-WROOM-32 s piny [6]

### 4.2 Modul SD karty

Tento modul zprostředkovává hardwarovou komunikaci s microSD kartou na Obr. 3, na které jsou uložena data pro řízení akvarijního zařízení a také na SD kartu ukládá získané údaje.

Modul kromě konektoru pro vložení SD karty obsahuje též převodník vstupního napětí z 5 V na pracovní napětí 3,3 V. Proto je nutné napájecí pin připojit na 5 V a ne na 3,3 V, na kterém pracuje mikrokontroler ESP-32. Komunikace s ESP32 probíhá po sběrnici SPI. Vstupy a výstupy na modulu SD karty z pohledu zdola na Obr. 4 jsou vypsány v Tab. 1.



Obr. 3: Modul SD karty (pohled shora)





Obr. 4: Modul SD karty (pohled zdola)

Tab. 1: Vstupy a výstupy modulu SD karty

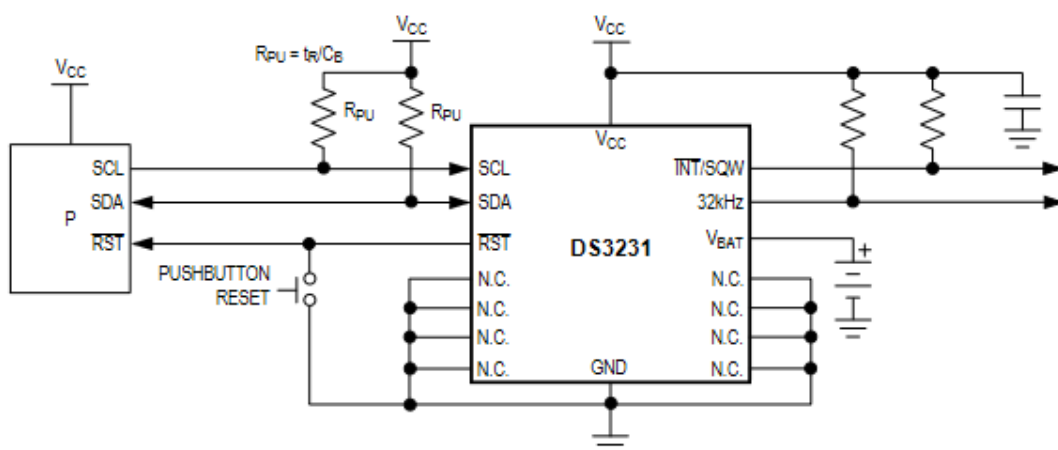
Vstupy a Výstupy	Význam
CS	Výběr zařízení
SCK	Hodinový signál
MOSI	Výstup dat (master)
MISO	Vstup dat (master)
VCC	Napájení 5 V
GND	Zem

### 4.3 Modul DS3231 + paměť EEPROM AT24C32

Tento modul obsahuje obvod reálného času DS3231 od firmy Maximintegrated. Na Obr. 5 je doporučené zapojení DS3231. Registry pro nastavení DS3231 jsou v Tab. 2. V našem případě jsou použity pouze registry pro nastavení a čtení času a datumu. Modul dále obsahuje paměť Flash AT24C32, která se dá použít pro ukládání dat. V této aplikaci nebude použita.

Mezi jeho vlastnosti patří [7]:

- Hodiny v reálném čase počítají sekundy, minuty, hodiny, datum měsíce, měsíc, den v týdnu a rok, s kompenzací přestupného roku platnou až do 2100.
- Přesnost  $\pm 2$  ppm v rozsahu teplot od  $0\text{ }^{\circ}\text{C}$  do  $+40\text{ }^{\circ}\text{C}$ .
- Přesnost  $\pm 3,5$  ppm v rozsahu teplot od  $-40\text{ }^{\circ}\text{C}$  do  $+85\text{ }^{\circ}\text{C}$ .
- Výstup digitálního teplotního senzoru: Přesnost  $\pm 3\text{ }^{\circ}\text{C}$ .
- Výstup RST / Tlačítko Reset Debounce Input.
- Dva denní alarmy.
- Programovatelný výstupní signál obdélníkových vln.
- Rychlé rozhraní I2C (400 kHz).
- Vstup pro záložní baterii pro nepřetržité měření času.
- Provoz s nízkou spotřebou prodlužuje dobu provozu na záložní baterii.
- Napájecí napětí v rozmezí od 2,3 V do 5,5 V.



Obr. 5: Doporučené zapojení DS3231 [7]

Tab. 2: Registry pro nastavení DS3231 [7]

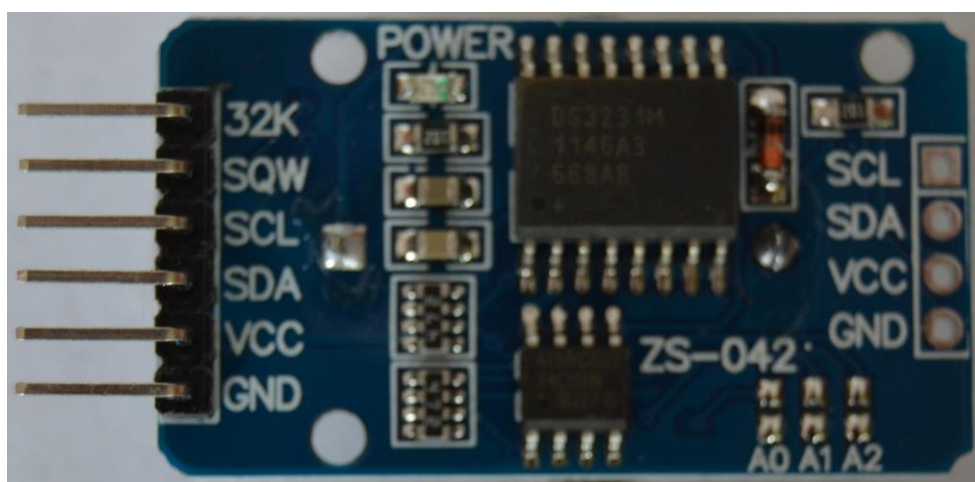
ADDRESS	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 LSB	FUNCTION	RANGE
00h	0	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12/24	AM/PM 20 Hour	10 Hour	Hour				Hours	1–12 + AM/PM 00–23
03h	0	0	0	0	0	Day			Day	1–7
04h	0	0	10 Date			Date			Date	01–31
05h	Century	0	0	10 Month	Month				Month/ Century	01–12 + Century
06h	10 Year				Year				Year	00–99
07h	A1M1	10 Seconds			Seconds				Alarm 1 Seconds	00–59
08h	A1M2	10 Minutes			Minutes				Alarm 1 Minutes	00–59
09h	A1M3	12/24	AM/PM 20 Hour	10 Hour	Hour				Alarm 1 Hours	1–12 + AM/PM 00–23
0Ah	A1M4	DY/DT	10 Date			Day			Alarm 1 Day	1–7
0Bh	A2M2	10 Minutes			Date				Alarm 1 Date	1–31
0Ch	A2M3	10 Minutes			Minutes				Alarm 2 Minutes	00–59
0Dh	A2M4	12/24	AM/PM 20 Hour	10 Hour	Hour				Alarm 2 Hours	1–12 + AM/PM 00–23
0Eh	A2M4	DY/DT	10 Date			Day			Alarm 2 Day	1–7
0Fh	EOSC	BBSQW	CONV	RS2	RS1	INTCN	A2IE	A1IE	Control	—
10h	OSF	0	0	0	EN32kHz	BSY	A2F	A1F	Control/Status	—
11h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	Aging Offset	—
12h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	MSB of Temp	—
13h	DATA	DATA	0	0	0	0	0	0	LSB of Temp	—

Pro zálohu dat při odpojení napájení je možno vložit nabíjecí baterii LIR2032 3,6 V. Tato baterie je v modulu po připojení napájecího napětí dobíjena. Místo baterie LIR2032 lze použít i klasickou knoflíkovou baterii CR2032 dle Obr. 6. Tato baterie není nabíjecí, a proto je vhodné odpájet rezistor 200 $\Omega$  na desce, aby se zamezilo nabíjení.

Modul reálného času z pohledu shora je na Obr. 7 a jeho vstupy a výstupy na obou koncích jsou uvedeny v Tab. 3. Výchozí I2C adresa je 0×57 a lze ji v případě potřeby změnit pomocí pájecích bodů A0-A1-A2.



Obr. 6: Zadní strana modulu reálného času s vloženou baterií



Obr. 7: Modul Reálného času (RTC)

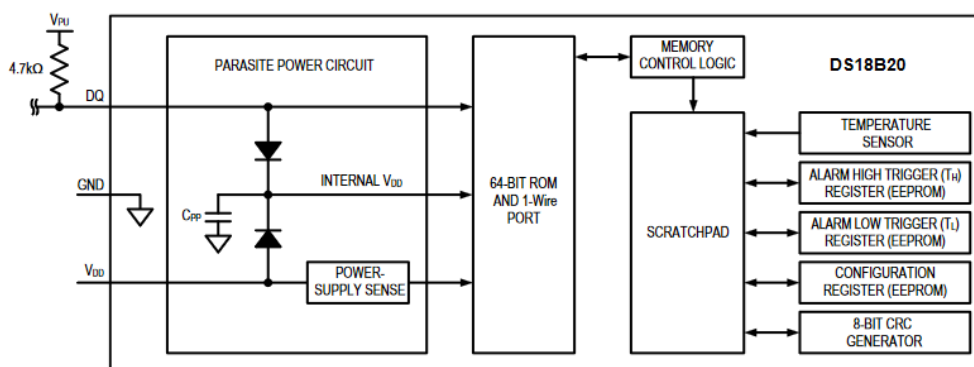
Tab. 3: Vstupy a výstupy RTC [7]

	Vstupy a výstupy	Význam
Levá strana	32k	Výstup frekvence vnitřního oscilátoru 32.768kHz
	SQW	Výstup obdélníkového signálu
	SCL	I2C Vstup hodin (Serial Clock Input)
	SDA	I2C Vstup/ výstup dat (Serial Data Input/Output.)
	VCC	Napájení od 2,3 do 5,5 V
	GND	Zem
Pravá strana (slouží k připojení dalšího zařízení)	SCL	I2C Vstup hodin (Serial Clock Input)
	SDA	I2C Vstup/ výstup dat (Serial Data Input/Output.)
	VCC	Napájení od 2,3 do 5,5 V
	GND	Zem

#### 4.4 Teplotní čidla DS18B20 od firmy Maximintegrated

Digitální teploměr DS18B20 umožňuje měření teploty v rozlišení od 9 do 12 bitů. Má funkci alarmu s energeticky nezávislými, uživatelsky programovatelnými horními a dolními spouštěcími body. DS18B20 komunikuje po sběrnici 1-Wire, která ze své podstaty vyžaduje pro komunikaci s mikroprocesorem pouze jednu datovou linku (a zem). Kromě toho může DS18B20 získávat energii přímo z datové linky („parazitní napájení“), což eliminuje potřebu externího napájení. Každý DS18B20 má jedinečný 64bitový sériový kód, který umožňuje fungování více DS18B20 na stejné 1-Wire sběrnici. Je tedy jednoduché použít jeden mikroprocesor k ovládání mnoha DS18B20 distribuovaných na velké ploše.

Mezi aplikace, které mohou těžit z této funkce, jsou systémy pro monitorování teploty uvnitř budov, zařízení nebo strojů a systémy pro monitorování a řízení procesů. Na Obr. 8 je blokové schéma vnitřního zapojení teplotního čidla [8].



Obr. 8: Zapojení čidla [8]

Specifikace [8]:

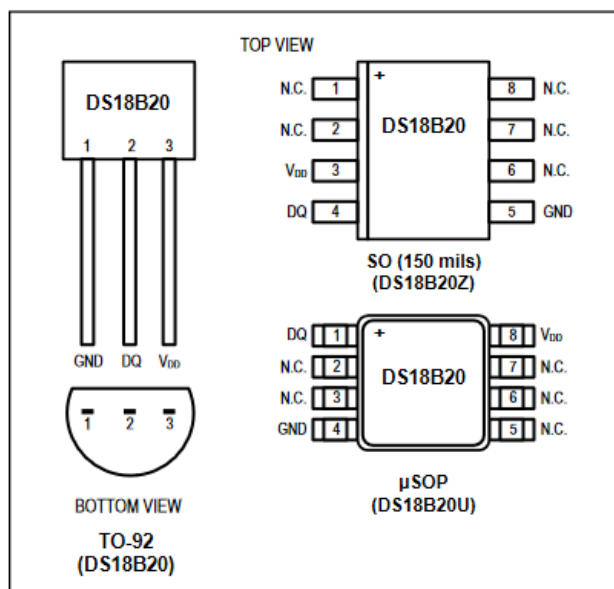
- Jednoduché rozhraní 1-Wire vyžaduje pro komunikaci pouze jeden pin.
- Rozsah teplot od -55 °C do + 125 °C (-67 °F do + 257 °F).
- Přesnost v rozsahu zobrazuje viz Tab. 4.
- Programovatelné rozlišení od 9 bitů do 12 bitů je v Tab. 5. Po zapnutí je rozlišení nastaveno na 12-bitů.
- Napájecí napětí je v rozmezí od 3,0 V do 5,5 V.
- Uživatelsky definovatelné nastavení alarmu s příkazem pro vyhledávání alarmů (identifikuje zařízení s teplotami mimo naprogramované limity.).
- Režim parazitního napájení vyžaduje pro provoz pouze 2 vodiče (DQ a GND).
- Každé zařízení má v paměti ROM uložen jedinečný 64bitový sériový kód.
- Dostupné v 8 pin SO 8 pin μSOP a 3 pin pouzdro TO-92 lze vidět na Obr. 9.

Tab. 4: Přesnosti v rozsahu teplotních čidel:

Od	Do	Přesnost
-10 °C	+ 85 °C	± 0,5 °C
-30 °C	+100 °C	± 1 °C
-55 °C	+125 °C	± 2 °C

Tab. 5: Rozlišení DS18B20

Rozlišení		Max. doba konverze [ms]
9 bit	0.5 °C	93,75
10 bit	0.25 °C	187,5
11 bit	0.125 °C	375
12 bit	0.0625 °C	750



Obr. 9: Typy pouzder DS18B20 [8]

Čidla pro měření teploty byla použita ve vodotěsném provedení na Obr. 10. To se skládá ze standardního provedení čidla DS18S20 v pouzdro TO-92 připojeného třívodičovému kabelu o délce 1 metr. Samotné čidlo spolu s kabelem je zalito v nerezové dutince, jejíž konec a část kabelu je opatřen smršťovací bužírkou k zamezení průniku vody.



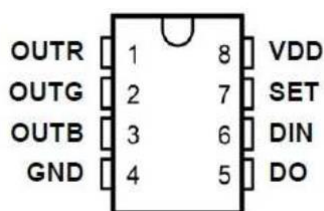
Obr. 10: Teplotní čidlo DS18B20 vodotěsné provedení

## 4.5 Modul PWM ovládání Led pásků

Tento modul je vlastnoruční dělaná konstrukce, založená na PWM ovladači LED WS2811, oddělovacích optočlenech a výkonových tranzistorech. Na Obr. 12 je hotová a osazená prototypová deska.

WS2811 parametry:

- Má 3 výstupní kanály se speciálními budicími obvody pro LED viz Obr. 11. Každý výstup může dosáhnout až 256 hodnot zobrazení jasu, což při základních barvách RGB odpovídá 167 777 216 barevných odstínů.
- Rychlost přenosu dat je nastavitelná na 400 Hz nebo 800 Hz pomocí vstupu SET. Vzdálenost mezi dvěma obvody může být až 5 metrů bez vlivu na rychlost a kvalitu spojení.
- Integrovaný obvod je interně vybaven inteligentním digitálním portem a zesilovacím obvodem signálu. Zahrnuje také přesný interní oscilátor a programovatelný výstup s konstantním proudem na napětí 12 V. Za účelem snížení zvlnění napájení jsou 3 výstupní kanály navrženy tak, aby zpozdily funkci zapnutí.
- Po resetování čipu po zapnutí přijme port DIN data z řadiče, první odvod přijímá počáteční 24bitová data, která se poté odešlou do interního datového registru, a ostatní data, která se upraví pomocí zesilovacího obvodu pro tvarování interního signálu, zasílají do další kaskády přes DO port. Do doby, než čip přijme nová data, vysílá PWM signály do výstupů R G B.
- Stručný popis jednotlivých vstupů výstupů modulu PWM je v Tab. 6.



Obr. 11: Vstupy a Výstupy modulu PWM [10]



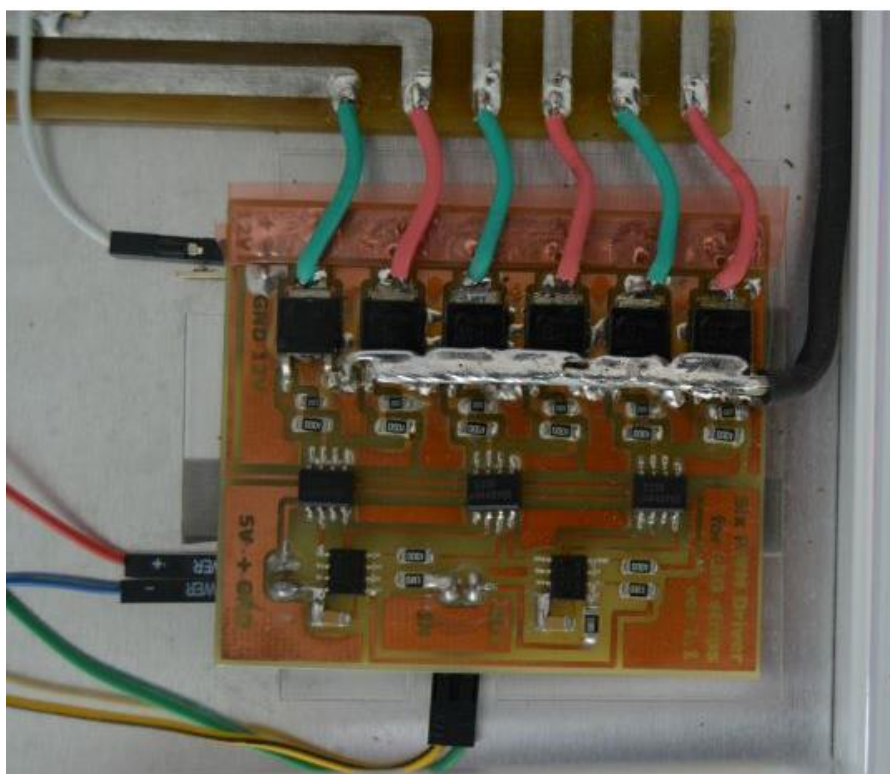
Tab. 6: Stručný popis vstupů a výstupů modulu PWM

VDD	Od 4,5 do 5,5V
GND	Zem
OUTR, G, B	PWM výstupy na LED (červená, zelená, modrá)
DIN	Vstup dat
DO	Výstup dat na další obvod
SET	Nastavení přenosové rychlosti LS (nízká rychlost) připojit na VDD
	Nastavení přenosové rychlosti HS (vysoká rychlost) nezapojen

Stručný popis funkce:

Data přijatá z mikroprocesoru se uloží do dvou IO WS2811. Jejich PWM výstupy budí Led diody v optočlenech OP1 až OP6. Tranzistory na jejich výstupech ovládají výkonové tranzistory Q1 až Q6.

Tyto tranzistory pracují v režimu spínačů a zesilují budící signály z optočlenů. Na jejich výstupy jsou připojeny Led pásy pro osvětlení akvária. Zároveň optočleny oddělují mikroprocesor ESP32 od výkonové části napájené napětím 12 V. Vstupy IO WS2811 pracují s logickou úrovní 5 V. ESP-32 naproti tomu pracuje s logickou úrovní 3,3V. Je proto žádoucí provést korekci logických úrovní. Existují na to speciální IO např. 74HCT125 nebo již hotové moduly např. SparkFun Logic Level Converter. V našem případě jde pouze o komunikaci jedním směrem tedy z mikroprocesoru do modulu, a tudíž si vystačíme s tranzistorem PNP v roli spínače.



Obr. 12: Modul WS2811

## 4.6 Modul Relé

Modul byl zakoupen jako hotový celek. Výkonová část se skládá ze 4 relátek určených pro zapínání a vypínání akvaristické techniky, jak je vidět na Obr. 13.

Parametry relé pro maximální spínací napětí a proud:

- 250 V AC - 10 A
- 30 V DC - 10 A

Napětí, při kterém relé sepne je 5 V DC a proud při sepnutí 70 mA. Jednotlivá relé jsou ovládána tranzistory. Mezi emitor a kolektor jednotlivých tranzistorů jsou antiparalelně zapojeny diody, která chrání tranzistory před napěťovými špičkami vznikajícími při spínání indukční zátěží. Báze tranzistorů jsou ovládány výstupy z optočlenů. Vstupy optočlenů jsou vyvedeny na konektor vstupů. Do série se vstupy optočlenů jsou připojeny LED diody, které opticky indikují zapnutí jednotlivých relé. Pro zvýšenou bezpečnost a stabilitu napájení mikroprocesoru je možno propojkou oddělit 5 V napájecí část relé od napájení mikroprocesoru. Logická hodnota pro spínání relé je 0, proto byly na vstupy připojeny spínací tranzistory pro změnu logické hodnoty na kladnou. Spínací hodnoty jsou vyčteny z pouzder relé a popisu na PCB. Proud protékající sepnutým relé byl zjištěn měřením.



Obr. 13: Modul Relé

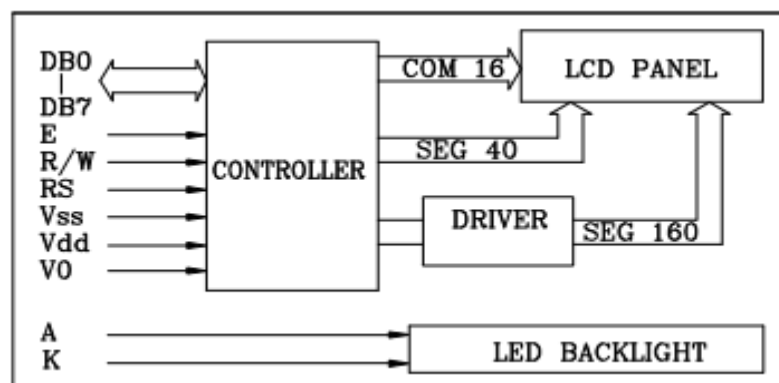


## 4.7 Stavový LCD displej

Slouží k zobrazení základních hodnot řídicí jednotky OTO. Základní schéma zapojení vnitřní části displeje je na Obr. 14. Řadič displeje je založen na čipové sadě Hitachi HD44780 (nebo kompatibilní). Vstupy a výstupy displeje jsou v Tab. 7.

Tab. 7: Vstupy a výstupy LCD displeje

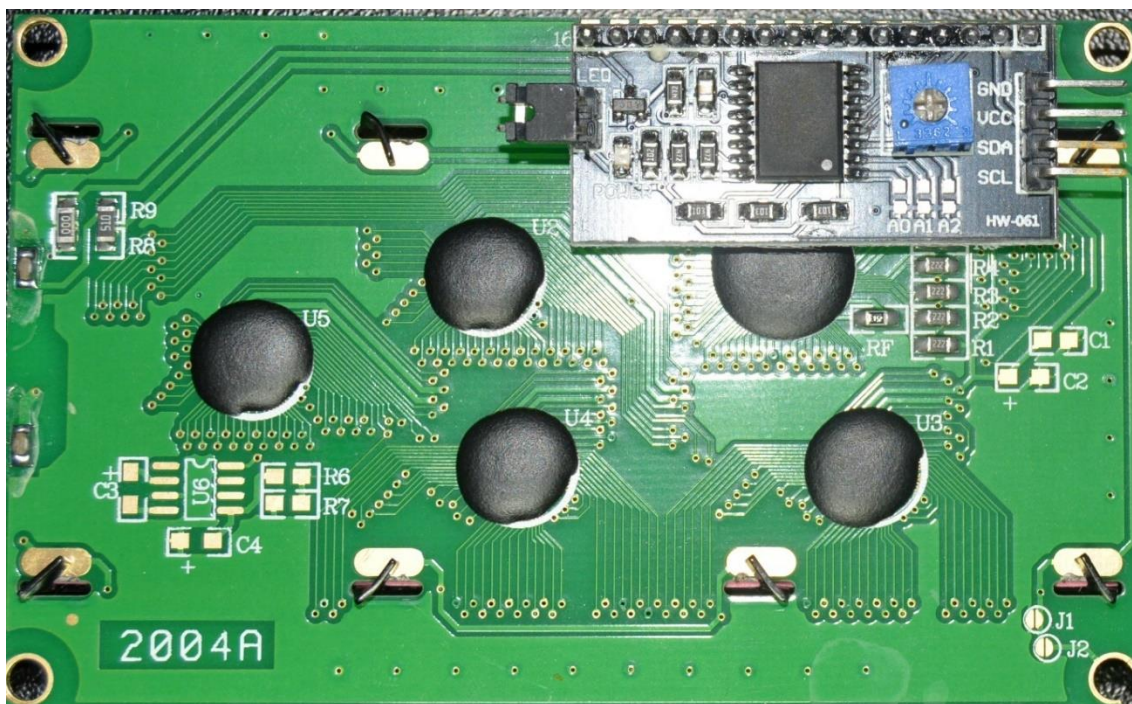
DB0 až DB7	8bitová sběrnice pro přenos dat.
E enable	Slouží k povolení signálu pro zápis nebo čtení dat.
R/W	Přepínání módů: 1–čtení 0–zápis
RS	Přepínání mezi datovým a instrukčním registrem
VSS	Zem
VDD	Napájecí napětí max 7 V
V0	Vstup pro řízení kontrastu
A	Napájení podsvícení
K	Zem podsvícení



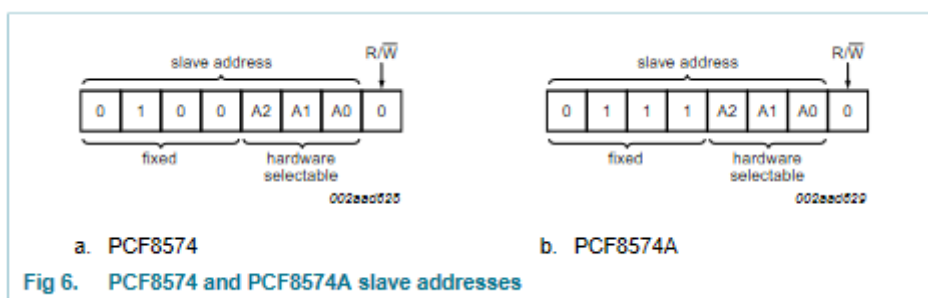
Obr. 14: Blokový diagram LCD displeje [11]

Existují dva typy. První z nich nemá pod displejem reflexní vrstvu, a tudíž potřebuje posvícení i ve dne. Pozadí displeje je modré a písmo bílé. Druhý má pod displejem reflexní vrstvu, a proto je za světla čitelný i bez podsvícení. To je potřeba pouze při slabém okolním světle. Pozadí má nazelenalou barvu a text je černý. Počet řádků jsou 4 a počet sloupců je 20. Počet bodů na znak je  $5 \times 7$  + kurzor.

Displej se dá koupit i v provedení pro sběrnici I2C a napájecí napětí 3,3 V. Tento typ displeje na Obr. 15 je použit v našem projektu. Výhody tohoto provedení spočívají v úspoře datových a ovládacích linek, kdy se z 12 zmenší pouze na 2. Pokud displej obsahuje převodník z I2C na 8 bit PCF8574 od firmy Texas, je standardní adresa displeje I2C  $0 \times 27$ . Změnu adresy je možno provést pomocí tří zkratovacích plošek na PCB A0–A1–A2 v rozmezí  $0 \times 20$ – $0 \times 27$  viz Obr. 16.



Obr. 15: Displej s I2C



Obr. 16: Registr adres PCF8574 [12]

## 4.8 Napájecí zdroj

Pro napájení byl použit zdroj z vyřazeného PC, který svými parametry bohatě splňuje požadované hodnoty napětí a proudu řídicí jednotky OTO.

Výkonové parametry zdroje:

- Celkový maximální výkon zdroje je 350 W.
- Napětí 3,3 V, max. proud 21 A.
- Napětí 5 V, max. proud 17 A.
- Napětí 12 V, max. proud 30 A.

Požadavky pro řídicí jednotku OTO:

- Napětí 5 V, max. proud 300 mA.
- Napětí 3,3 V, max. proud 1,2 A.
- Napětí 12 V, max. proud podle délky a výkonu LED pásků. V tomto zařízení byly použity pásy těchto parametrů:
  - Příkon 14,4 W/m.
  - 60 diod na 1 metr.
  - Typ diod 5050.
  - Délka pásu na osvětlení (1 kanál) je 3×60 cm. Při počtu šesti kanálů se jedná o celkovou délku všech pásků 10,8 m.

Celkový příkon je  $10,8 \times 14,4 = 156$  W. Při napájecím napětí 12 V je protékající proud při plném osvětlení cca 13 A.

## **4.9 Tlačítko**

Slouží pro zobrazení IP adresy řídicí jednotky OTO na stavovém displeji.

## 5 Programová část řídicího systému

V této kapitole bude probrána vlastní tvorba kódu i s vysvětlením funkcí jednotlivých celků. Kód je naprogramovaný v programu Visual Studio Code s rozšířením PlatformIO IDE. Byl použit programovací jazyk C.

### 5.1 Úvodní část programu

V úvodní části programu pomocí příkazu `#include` byly naimportovány potřebné knihovny podle Obr. 17. Některé knihovny nejsou přímo součástí prostředí PlatformIO IDE a je potřeba je dodatečně naimportovat.

- „<WiFi.h>“ umožňuje bezdrátové připojení k internetu. Slouží oběma směry (jak příchozí, tak i odchozí spojení) a podporuje protokoly WEP a WPA2 sloužící k šifrování (zabezpečení přenosu dat). Tato knihovna se automaticky nainstaluje při instalaci doplňku ESP 32 do PlatformIO IDE.
- „<ESPAsyncWebServer.h>“ je použitá knihovna k vytvoření a obsluze web serveru.
- „<AsyncTCP.h>“ slouží jako základ pro knihovnu „<ESPAsyncWebServer.h>“ (ta je vlastně nástavbou základní knihovny „<AsyncTCP.h>“), v kódu se s ní nepracuje (pracujeme s ní nepřímo, právě přes „<ESPAsyncWebServer.h>“).
- „<Arduino\_JSON.h>“ je základem formátu JSON (JavaScript Object Notation) – jedná se o zjednodušený formát pro výměnu dat, vhodný pro široké spektrum programovacích jazyků založených na architektuře jazyků C. V kódu je to použito k přenosu dat z ESP serveru klientovi na webovou stránku.
- „<OneWire.h>“ *„1-Wire je sběrnice navržena firmou Dallas Semiconductor Corp. pro komunikaci zařízení nízkou datovou rychlostí, signalizaci i napájení. 1-Wire je podobná ke sběrnici I<sup>2</sup>C, jen s nižší datovou propustností a delším dosahem. Obvykle je používána pro komunikaci s malými levnými zařízeními jako jsou termometry a další různá zařízení. Síť z 1-Wire zařízení s připojeným master zařízením se nazývá MicroLAN [9].“*
- „<DallasTemperature.h>“ slouží pro komunikaci s teplotními čidly DS18B20. Jedná se o nástavbu ke knihovně „<OneWire.h>“.
- „<time.h>“ je standardní knihovna jazyka C pro práci s časem a datem.
- „SD.h“ slouží ke komunikaci modulu SD karty s ESP32. Ke své činnosti potřebuje následující knihovny „SPI.h“ a „FS.h“.
- „Freenove\_WS2812\_Lib\_for\_ESP32.h“ slouží pro řízení čipů WS2812, které podle přijatých dat pomocí signálů PWM řídí jas jednotlivých LED čipů. Pro každý LED modul je třeba poslat 3×8 bitů (hodnoty 0-255).
- „<LiquidCrystal\_I2C.h>“ slouží ke komunikaci s LCD displejem pomocí sběrnice I2C.

```

36 // WiFi
37 #include <WiFi.h> // https://www.arduino.cc/en/Reference/WiFi
38
39
40 #include <AsyncTCP.h> // https://github.com/me-no-dev/AsyncTCP
41 #include <ESPAsyncWebServer.h> // https://github.com/me-no-dev/ESPAsyncWebServer#the-async-web-server
42 #include <Arduino_JSON.h> // https://github.com/arduino-libraries/Arduino\_JSON
43 // #include "AsyncJson.h"
44
45 // Teplotní čidla DS18B20.
46 #include <Wire.h>
47 #include <OneWire.h> // https://www.pjrc.com/teensy/td\_libs\_OneWire.html
48 #include <DallasTemperature.h> // Knihovna pro teplotní čidlo DS18B20. https://github.com/milesburton/Arduino-Temperature-Control-Library
49
50 // Knihovna pro práci s časem
51 #include <time.h> // https://www.tutorialspoint.com/c\_standard\_library/time\_h.htm
52
53 // SD karta
54 #include "FS.h"
55 #include "SD.h" // https://github.com/espressif/arduino-esp32/tree/master/libraries/SD
56 #include "SPI.h"
57
58 // WS2811.
59 #include "Freenove_WS2812_Lib_for_ESP32.h" // https://github.com/Freenove/Freenove\_WS2812\_Lib\_for\_ESP32
60
61 // LCD displej
62 #include <LiquidCrystal_I2C.h> // https://github.com/johnrickman/LiquidCrystal\_I2C
63 LiquidCrystal_I2C lcd(0x27, 20, 4); // Nastavení LCD adresy na 0x27. Displej má 20 znaků a 4 řádky.
64

```

Obr. 17: Použité knihovny s odkazy

## 5.2 Počáteční nastavení a inicializace jednotlivých periférií

### 5.2.1 Teplotní čidla DS18B20

Aplikace obsahuje 3 teplotní čidla, která jsou použita pro měření teploty. První dvě čidla jsou umístěna v akváriu na různých místech a z jejich teplot se vypočítává průměrná teplota vody. Tato hodnota se dále porovnává s povolenou teplotou vody a tento výsledek je použit pro zapínání a vypínání topícího tělesa, které ohřívá vodu. Aby bylo možné komunikovat s teplotními čidly a získávat z nich naměřené hodnoty teplot, je potřeba znát jejich 64bitovou sériovou komunikační adresu. Pro nastavení teplotních čidel je využit příkaz „begin()“ na aktivaci čidel. Dále příkazem „setResolution ([ID čidla],11)“ je nastaveno rozlišení měřené teploty na 11 bitů, což znamená, že vrácená hodnota bude mít rozlišení 1/8 °C. Příkazem „setWaitForConversion(false)“ bude nastavena konverze teploty na digitální hodnotu bez čekání na výsledek (ten se vyzvedne později). Příkazem „requestTemperatures()“ tento převod bude spuštěn.

### 5.2.2 Osvětlení

Je tvořeno šesti kanály z LED pásků, které jsou ovládány pomocí dvou WS2811. Ovládání probíhá přes knihovnu „Freenove\_WS2812\_Lib\_for\_ESP32.h“. Na začátku je potřeba smazat registry v obvodech WS2812, čímž se nastaví výstupy na 0 a všechny LED pásy zhasnou. To se provede následující sekvencí příkazů, která ukazuje Obr. 18:

- Příkaz „begin()“ spustí komunikaci s WS2811.
- Příkazem „strip.setBrightness(255)“ nastaví na maximální hodnotu.
- Následující smyčka projde jednotlivé obvody WS2811, v našem případě 2, a nastaví výstupy na 0.
- Příkazem „strip.show()“ provede zápis hodnot z vnitřní paměti WS2811 na výstupy. Tím je zaručeno, že po zapnutí řídicí jednotky pro ovládání teploty vody a osvětlení v akváriu, dále jen řídicí jednotky OTO, bude osvětlení vypnuto.

```

244 // Nastavení LED osvětlení Ws2811.
245 strip.begin();
246 strip.setBrightness(255);
247 for (uint8_t ch = 0; ch < pocet_ch; ch++) strip.setLedColorData(ch, 0, 0, 0);
248 strip.show();

```

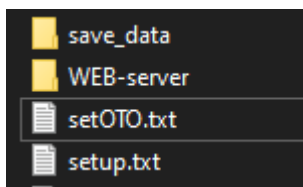
Obr. 18: Nastavení LED osvětlení

### 5.2.3 SD karta

Slouží k následujícím účelům:

- Po zapnutí se ze souboru setup.txt, získají přihlašovací údaje pro připojení k místní WIFI síti (název sítě a heslo).
- Ze souboru setOTO.txt se každou hodinu načtou nová data pro nastavení jednotlivých kanálů osvětlení, pro požadovanou teplotu vody a pro nastavení výstupů, topení, UV sterilizéru a vzduchování.
- Při změně hodnot osvětlení, teploty nebo výstupů a také v pravidelných intervalech 10 s dojde k zápisu těchto hodnot i s časovým razítkem do souboru s názvem YYYYMMDD.csv
- (kde Y – rok, M – měsíc a D – je den). Tyto soubory se vytváří pro každý den a jsou uloženy v adresáři save\_data.
- Na kartě se dále nachází HTML stránky, které slouží pro ovládání a komunikaci s webovým serverem ze strany klienta, nastavení OTO a ukládání datových souborů na zařízení klienta

Následující struktura souboru a jejich umístění na SD kartě musí být zachována, jinak řídicí jednotka OTO nebude fungovat viz Obr. 19.



Obr. 19: Struktura souboru

Popis jednotlivých složek a souborů:

- save\_data - obsahuje soubory s údaji o intenzitě osvětlení jednotlivých LED kanálů, teploty vody, stavu výstupů a časového údaje, kdy byly tyto data pořízena. Každý soubor obsahuje hodnoty z jednoho dne.
- WEB-server - zde jsou HTML stránky a ikona faviconu pro obsluhu OTO pomocí webového rozhraní.
  - Index.html - hlavní stránka OTO.
  - set\_oto.html - slouží pro úpravu řídicího souboru OTO.
  - dataset.html - zde je možné uložit naměřená data pro další zpracování např. grafy.
  - notfound.html - toto se zobrazí při pokusu připojení na neexistující stránku.
- setOTO.txt - soubor s nastavením řídicí jednotky.
- setup.txt - soubor s údaji pro připojení k WIFI.

## 5.2.4 Nastavení výstupů pro ovládání topení, UV sterilizéru a vzduchování

Popisy obrázků:

- Příkazem `#define` jsou přiřazeny názvy výstupů konkrétním pinům na ESP 32 na Obr. 20.
- Příkazem „pinMode“ jsou nastaveny jednotlivé piny jako výstupy což ukazuje Obr. 21.
- Příkazem „digitalWrite“ tyto výstupy budou nastaveny na počáteční hodnoty.
- Výstupní piny jsou připojeny na reléový modul, kde přes optočleny spínají výkonové relé.

Díky optočlenům jsou výstupy ESP 32 galvanicky odděleny od síťového napětí, a tudíž nehrozí jeho zničení.

```
143 // Vystupy pro ovladani topeni, sterilizeru a vzduchovani.
144 #define out_topeni 25
145 uint8_t rele_out_topeni = 0;
146
147 #define out_sterilizer 26
148 #define out_vzduchovani 27
149
```

Obr. 20: Definované výstupy T-U-V

```
262 // Nastaveni vystupu.
263 pinMode(out_topeni, OUTPUT);    digitalWrite(out_topeni, LOW);
264 pinMode(out_sterilizer, OUTPUT); digitalWrite(out_sterilizer, LOW);
265 pinMode(out_vzduchovani, OUTPUT); digitalWrite(out_vzduchovani, LOW);
266
```

Obr. 21: Nastavení výstupů pro T-U-V

## 5.2.5 Spuštění a konfigurace sítě WIFI

ESP 32 může fungovat jako WIFI stanice, přístupový bod nebo obojí.

- WIFI\_STA - Režim stanice ESP 32 se připojuje k přístupovému bodu.
- WIFI\_AP - Režim přístupového bodu, stanice se mohou připojit k ESP 32.
- WIFI\_STA\_AP - Smíšený režim, ESP 32 funguje jak v režimu stanice, tak přístupového bodu.

Protože budeme ESP 32 připojovat k našemu bezdrátovému routeru, bude zvolen režim WIFI\_STA viz Obr. 22. WIFI síť bude spuštěna příkazem „begin()“ a parametry zadány jako název sítě a heslo. Ve smyčce je kontrolován „WiFi.status()“ dokud se nerovná WL\_CONNECTED, což znamená úspěšné připojení k síti. Další návratové hodnoty, které se dají kontrolovat u příkazu „WiFi.status()“, jsou v Tab. 8.



Tab. 8:Návratové hodnoty sítě WIFI

Hodnota	Konstanta	Význam
0	WL_IDLE_STATUS	Stav po příkazu WiFi.begin()
1	WL_NO_SSID_AVAIL	Není k dispozici žádný název sítě
2	WL_SCAN_COMPLETED	Skenování sítě dokončeno
3	WL_CONNECTED	Připojeno k síti WIFI
4	WL_CONNECT_FAILED	Připojení k síti selhalo
5	WL_CONNECTION_LOST	Připojení k síti bylo ztraceno
6	WL_DISCONNECTED	Odpojení od sítě

```

516 // Spustení WiFi.
517 void start_WiFi() {
518
519     WiFi.mode(WIFI_STA);
520
521     Serial.printf("\n\nPřipojuji se k:%s.\n", ssid);
522     Serial.printf("Heslo k wifi je:%s.\n", password);
523     WiFi.begin(ssid, password);
524
525     lcd.setCursor(0, 2); // Sloupec, radek.
526     lcd.print("wifi");
527     uint8_t time_out = 0;
528     uint8_t time_poz = 4;
529     while (WiFi.status() != WL_CONNECTED) {
530         delay(500);
531         Serial.print(".");
532
533         if (!(time_out % 4)) {
534             lcd.setCursor(time_poz, 2); lcd.print(".");
535             time_poz++;
536         }
537         if (time_poz > 18) {
538             time_out = 0;
539             time_poz = 4;
540             lcd.setCursor(time_poz, 2); lcd.print("
541         }
542         time_out++;
543     }
544     lcd.setCursor(18, 2); lcd.print("ok");
545     Serial.println("WiFi připojena.");
546     IPAdr = String(WiFi.localIP().toString().c_str());
547     WiFiCh = String(WiFi.channel());
548     Serial.printf("IP adresa zařízení je: %s přideleny kanal: %s.\n", IPAdr.c_str(), WiFiCh.c_str());
549
550 }

```

Obr. 22: Spuštění WIFI

Po připojení k síti se do proměnné IPAdr vloží IP adresa řídicí jednotky OTO. Ta bude sloužit k zobrazení na displeji pro usnadnění připojení z webového prohlížeče. Do proměnné WiFiCh se uloží číslo přiděleného kanálu.



### 5.2.6 Nastavení času

Nastavení času probíhá při zapnutí řídicí jednotky OTO a dále periodicky každý měsíc.

- Zkontroluje se, zda řídicí jednotka OTO je připojena k síti WIFI.
- Příkazem „configTime(3600 \* timezone, daysavetime \* 3600, "ntp.globe.cz", "0.pool.ntp.org", "1.pool.ntp.org");“ se kontaktuje časový server NTP a pokusí se získat platný časový údaj.
- Příkazem „getLocalTime(&time\_ESP)“ se zkontroluje, zda byl předchozí příkaz úspěšný.

Pokud příkaz nebyl úspěšný, systémový čas je aktualizován z externího RTC modulu DS3231 a tím je nastavení času ESP ukončeno.

Pokud ano, načtou se jednotlivé časové údaje jako je hodina, minuta, vteřina, den, měsíc a rok do pole, které bude použito pro nastavení externího RTC modulu DS3231.

Hodnoty do jednotlivých registrů DS3231 budou zapsány příkazem „getLocalTime(&time\_ESP)“. Po zapsání všech hodnot do registrů se uložené hodnoty zpětně načtou do proměnné „time\_ESP“ a porovnají se s předchozími hodnotami. Pokud jsou stejné, aktualizuje se systémový čas.

### 5.2.7 Zobrazované údaje na stavovém displeji

Jedná se o znakový LCD s modrým podkladem a bílým písmem. Jelikož má pouze čtyři řádky po 20 znacích, neoplývá zrovna velkou informační schopností. Před vlastní inicializací je nutné pro zobrazení speciálních znaků (šipka nahoru, šipka dolů, 2 šipky proti sobě, teploty T1-T3 Obr. 23) definovat uživatelské symboly.



Obr. 23: LCD displej s ukázkou uživatelsky definovaných znaků

Ty se definují jako matice 8bitových znaků, kde každý prvek z matice představuje jeden bod znaku např. šipka nahoru se definuje takto:

- 0x04 00000100
- 0x0E 00001110
- 0x15 00010101
- 0x04 00000100
- 0x04 00000100
- 0x04 00000100
- 0x04 00000100
- 0x00 00000000

Z 8 bitů se použije pouze 5 prvních bitů, protože znak má rozměry 5×8 bodů. Z toho plyne tento zápis:

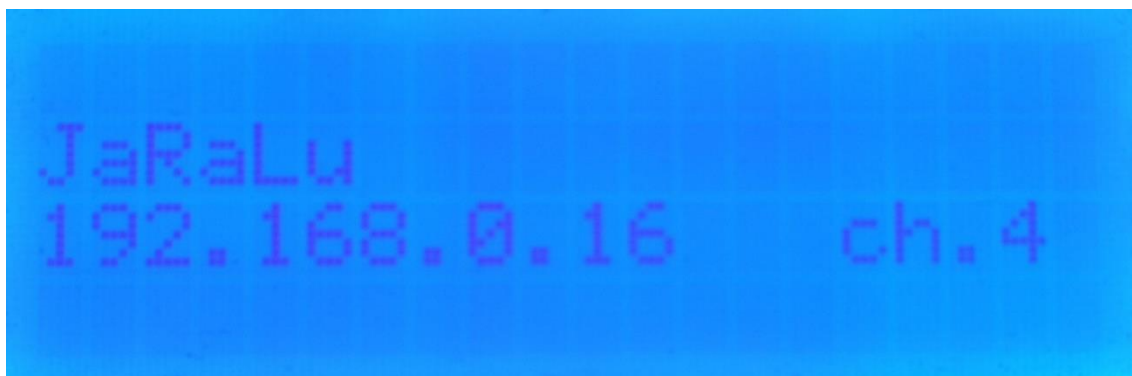
```
„uint8_t toparrow[8] = {0x04,0x0E,0x15,0x04,0x04,0x04,0x04, 0x00};“
```

U ostatních uživatelských znaků, které je potřeba vytvořit, je postup obdobný. Výsledek ukazuje Obr. 24. Následně je nutno v sekci „setup()“ spustit příkaz „createChar“, který přijímá 2 parametry. První z nich je pořadové číslo vytvořeného znaku a druhým je odkaz na proměnnou, která byla vytvořena výše. Tímto je umožněno používat znak na displeji pomocí příkazu „write“ (pořadové číslo znaku). Maximální počet uživatelsky definovaných znaků je 8.

```
61 // LCD displej
62 #include <LiquidCrystal_I2C.h> //https://github.com/johnrickman/LiquidCrystal_I2C
63 LiquidCrystal_I2C lcd(0x27, 20, 4); // Nastavení LCD adresy na 0x27. Displej má 20 znaků a 4 řádky.
64
65 // Definice uživatelských znaků na LCD.
66 uint8_t toparrow[8] = { 0x04,0x0E,0x15,0x04,0x04,0x04,0x04 };
67 uint8_t botarrow[8] = { 0x04,0x04,0x04,0x04,0x15,0x0E,0x04 };
68 uint8_t twoarrow[8] = { 0x11,0x0A,0x04,0x00,0x04,0x0A,0x11 };
69 uint8_t T1znak[8] = { 0x1F,0x04,0x04,0x04,0x04,0x04,0x14 };
70 uint8_t T2znak[8] = { 0x1F,0x04,0x04,0x04,0x14,0x04,0x14 };
71 uint8_t T3znak[8] = { 0x1F,0x04,0x14,0x04,0x14,0x04,0x14 };
```

Obr. 24: Knihovna LCD displeje a uživatelské znaky

Po zapnutí a připojení k Wi-Fi po dobu cca 10 s se zobrazuje název sítě a důležitou IP adresu řídicí jednotky OTO podle Obr. 25. Pomocí této adresy se přes webové rozhraní komunikuje s řídicí jednotkou OTO.



Obr. 25: Zobrazení informací o WIFI na stavovém displeji

Na Obr. 26 je standardní zobrazení stavového displeje, který ukazuje čas a datum, hodnotu osvětlení jednotlivých výstupů od 0 do 255 (0 - nesvítí, 255 - maximální svít), kde jednotlivé barvy LED pásků jsou označeny následujícími znaky:

- R-červená.
- G-zelená.
- B-modrá.
- W-teple bílá.
- U-ultra fialová.
- C-studeně bílá.

Za čísla hodnot jednotlivých barev se dále zobrazují tyto symboly:

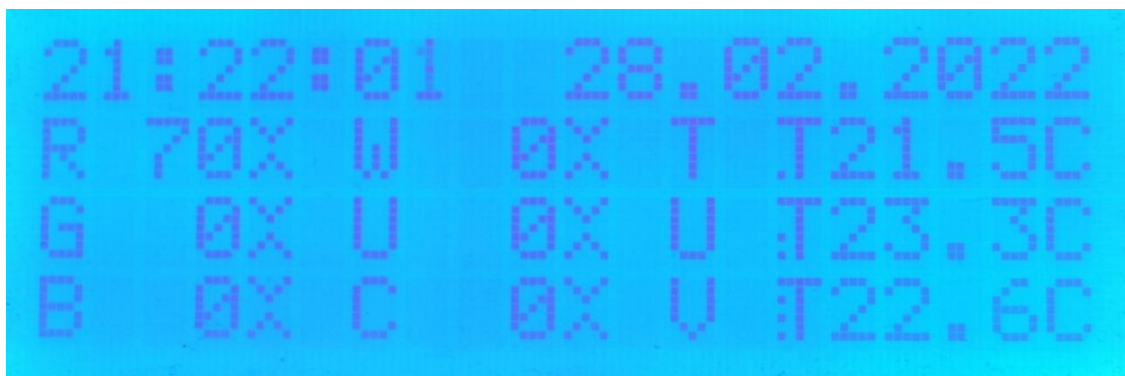
- šipka nahoru (hodnota nedosáhla hodnoty nastavené a stoupá)
- šipka dolů (hodnota je vyšší než nastavená a klesá)
- dvě šipky proti sobě (hodnota je stejná jako nastavená a již se nemění)

V pravé části displeje na řádcích 2-4 jsou vedle symbolů „T“ s jednou až třemi tečkami zobrazeny teploty z čidel t1-t3. Pokud se na displeji místo hodnoty teploty zobrazí „\*N-C\*“, znamená to, že se hodnota nepodařila získat.

Mezi hodnotami osvětlení a teploty, se ve sloupci zobrazují stavy jednotlivých výstupů.

- T-Topení
- U-UV sterilizer
- V-Vzduchování

Pokud je uvedený symbol zobrazen, znamená to, že výstup je zapnut a naopak.



Obr. 26: Standardní zobrazení stavového displeje

### 5.3 Web server

ESP 32 poskytuje asynchronní web server, což znamená, že běží nezávisle na hlavní smyčce programu, a tudíž není potřeba ho v této smyčce obsluhovat. Po instalaci nezbytných knihoven je potřeba vytvořit nový objekt web serveru, a to příkazem „AsyncWebServer server(80)“ na Obr. 27. Jako vstup je předán port, kde bude server poslouchat. Je použit port 80, což je výchozí port HTTP.

```

271 // Obsluha WEB Serveru.
272 // Poskytování souboru z vybraného adresáře.
273 // uri(url), datové medium (SD karta), adresář (na SD kartě).
274 server.serveStatic("/", SD, "/Web-server/");
275 server.serveStatic("/data", SD, "/save_data/");
276
277 // Server-Sent Events (SSE). Události odesílané z OTO k aktualizaci html stránek.
278 > events.onConnect([](AsyncEventSourceClient *client){ ...
282 });
283 server.addHandler(&events); // Přidání události.
284
285 // Pozadavek na zobrazení úvodní stránky OTO.
286 > server.on("/", HTTP_GET, [](AsyncWebServerRequest* request) { ...
289 });
290
291 // Navrácení hodnoty při změně parametru na Index.html. (Aktualizace OTO)
292 > server.on("/prenos", HTTP_POST, [](AsyncWebServerRequest* request) { ...
307 });
308 // Vyrizování jednotlivých požadavků od klienta.
309 > server.on("/setUp", HTTP_POST, [](AsyncWebServerRequest* request) { ...
407 });
408 server.onNotFound(notFound); // Pokud požadovaná stránka neexistuje, zobrazí se varování.
409 server.begin();
410

```

Obr. 27: Nastavení manipulačních cest

Pomocí příkazů „serveStatic“, které umožní sdílet vybraný adresář na zvoleném médiu, se zpřístupní adresáře na SD kartě, což později bude využito při ukládání souborů s daty a při zobrazení jednotlivých HTML stránek. Parametry tohoto příkazu jsou následující:

- cesta, která bude volána z prohlížeče
- typ media, na kterém se adresář nachází
- cesta k fyzickému umístění adresáře na zvoleném nosiči

Dále budou pro aktualizaci hlavní webové stránky použity události odeslané serverem (SSE) vyvolané příkazem „events.onConnect“ na Obr. 28. Pro odesílání hodnot klientovi je potřeba ještě nastavit zdroj události na serveru „server.addHandler(&events)“. SSE umožňuje odeslání aktualizovaných dat na HTML stránku bez žádání serveru o tyto data. Na straně HTML stránky dojde k jejich kontrole a poté jsou zobrazena v konkrétních elementech na stránce.

```

277 // Server-Sent Events (SSE). Události odesílané z OTO k aktualizaci html stránek.
278 events.onConnect([](AsyncEventSourceClient *client){
279     if(client->lastId()) {
280         Serial.printf("Client reconnected! Last message ID that it got is: %u\n", client->lastId());
281     }
282     // Posle událost: "Ahoj tady OTO", id počet milisekund
283     // a nastaví spoždění pro reconnect na 1 sekundu.
284     client->send("Ahoj tady OTO",NULL,millis(),1000);
285 });
286 server.addHandler(&events); // Přidání události.

```

Obr. 28: Události odeslané serverem

Dále pro vyřízení požadavků od klienta budou použity příkazy `server.on`. Tento příkaz vyžaduje následující 3 parametry:

- Nastavení odkazu na cestu, na které bude server naslouchat.
- Určuje typ požadavku, který bude server akceptovat (v našem případě jak `HTTP_POST` tak `HTTP_GET`).
- Jako třetí parametr bude aplikována funkce „send“ z třídy „AsyncWebServerRequest“, která obsluhuje požadavky klientů.

Příkaz „`server.on(“/“)`“ slouží k zobrazení HTML stránky Řídící jednotky OTO. Stačí do adresního řádku prohlížeče napsat IP adresu řídící jednotky dle Obr. 29. Tento znak „/“ značí root cestu WEB serveru, požadavek je typu `HTTP_GET` a jako třetí parametr je použita funkce „send“ z třídy „AsyncWebServerRequest“. Zde bude uvedeno typ souborového systému, umístění souboru, který bude požadován a typ tohoto souboru. Pro zobrazení souboru v prohlížeči bez uložení na disk bude vynechán 4. parametr nebo na toto místo vložena logická hodnota `false`.

```

285 // Požadavek na zobrazení úvodní stránky OTO.
286 server.on("/", HTTP_GET, [](AsyncWebServerRequest* request) {
287     screen = 1;
288     request->send(SD, "/Web-server/index.html", "text/html");
289 });
290
284 server.on("/", HTTP_GET, [](AsyncWebServerRequest* request) {
285     request->send(SD, "/Web-server/index.html", "text/html");
286 });

```

Obr. 29: Zobrazení HTML stránky

Příkaz „`server.on(“/prenos“)`“ slouží k přijetí dat, které byly odeslány klientem z web stránky viz Obr. 30.

Do řetězce „aktualizaceESP“ se uloží data což ukazuje Obr. 30, která se následně zpracují podprogramem „`proc_imp_msg()`“. Ten rozdělí řetězec na jednotlivé hodnoty podle oddělovače a ty jsou vloženy do proměnných pro nastavení periférií. Z webové stránky se takto ovládá stav výstupů a nastavení osvětlení akvária.

```

291 // Navratove hodnoty pri zmene parametru na Index.html. (Aktualizace OTO)
292 server.on("/prenos", HTTP_POST, [](AsyncWebServerRequest* request) {
293     if (request->hasParam(PARAM_INPUT)) {
294         String aktualizaceESP;
295         aktualizaceESP = request->getParam(PARAM_INPUT)->value();
296         Serial.printf("\naktualizaceESP - %s\n", aktualizaceESP.c_str());
297         if (aktualizaceESP.substring(0, 1) == "?") proc_imp_msg(aktualizaceESP); // Zpracuje vstupni retezec.
298         else if (aktualizaceESP == "screen") {
299             screen = 1;
300             screenOld = 1;
301         }
302         request->send(200, "text/plain", "OK");
303     } else {
304         Serial.println("Beze zmeny");
305         request->send(200, "text/plain", "OK");
306     }
307 });

```

Obr. 30: Aktualizace hodnot podle požadavků klienta

Příkaz „server.on(“/setUP“)” slouží k aktualizaci nastavení OTO a k ukládání souborů s naměřenými daty z karty SD na místní zařízení klienta na Obr. 31.

```

308 // Vyrizování jednotlivých požadavků od klientu.
309 server.on("/setUP", HTTP_POST, [(AsyncWebServerRequest* request) {
310     if(request->hasParam(PARAM_INPUT)){
311         String ret_res = "";
312         String transData = request->getParam(PARAM_INPUT)->value();
313         String headMsg = transData.substring(0, 7);
314 >         if(headMsg == "X?AgetA"){ // Zádost o data od klienta set_oto.html pro nastavení OTO. ...
315 >         }else if(headMsg == "X?AsetA"){ // Přijem dat od klienta set_oto.html pro nastavení OTO. ...
316 >         }else if(headMsg == "X?FgetL"){ // zádost o seznam datových souboru od klienta dataset.html. ...
317 >         }else if(headMsg == "X?FgetF"){ // zádost o stažení vybraného datového souboru od klienta dataset.html. ...
318 >         }else if(headMsg == "X?FdelF"){ // zádost o smazání staženého datového souboru od klienta dataset.html. ...
319 >         }else{ ...
320 >         if(ret_res != "") request->send(500, "text/plain", "Internal Server Error");
321 >         // Data pro SSE zpravy a serial.
322 >         Info_tisk("dataOTO",ret_res.c_str(),0);
323 >         ret_res = ""; // Blokace tisku.
324 >     }else{
325 >         Serial.println("Zadna zprava");
326 >         request->send(400, "text/plain", "Bad Request");
327 >     }
328 > });
329 > server.onNotFound(notFound); // Pokud požadována stránka neexistuje, zobrazí se varování.
330 > server.begin();
331 >

```

Obr. 31: Zpracování požadavků od klienta

Na cestě „/setUP“ přijme požadavek klienta a podle typu žádosti provede následující akci:

Žádost obsahuje řetězec:

- „X?AgetA“ - Zobrazení web stránky pro nastavení řídicí jednotky. Web server odešle obsah souboru setOTO.txt klientovi.
- „X?AsetA“ - Příjem aktualizovaných hodnot pro nastavení řídicí jednotky. Web server zkontroluje přijatá data a pokud jsou v pořádku, uloží je do souboru /upOTO.txt.
- „X?FgetL“ - Odeslání souboru, který obsahuje seznam všech datových souborů v adresáři „save\_data“ Pokud tento soubor neexistuje, tak bude vytvořen. To znamená, že se projde adresář „save\_data“ na SD kartě a názvy všech souborů s koncovkou .csv se do něho vloží. Poté je tento soubor odeslán klientovi.
- „X?FgetF“ - Odeslání souboru s požadovaným názvem.
- „X?FdelF“ - Smazání souboru, který obsahuje seznam datových souborů z důvodů jeho pozdější aktualizace.
- Jiné požadavky jsou ignorovány.

Poslední příkaz pro web server je „server.onNotFound(notFound)“, který při chybném dotazu přesměruje klienta na web stránku s upozorněním, že požadavek je chybný. Pro funkčnost je potřeba spustit server příkazem „server.begin()“.

Hlavní část programu probíhá ve smyčce, která se neustále opakuje. Nejprve se zkontroluje, zda přišli data z webu. Pokud ano, porovnají se přijatá data se stávajícími a pokud se některá liší, stávající data se upraví na nové hodnoty. Takto lze měnit intenzitu jednotlivých kanálů led osvětlení a výstupů. Tyto hodnoty jsou platné, dokud nedojde k jejich změně buďto z webového rozhraní anebo při načtení nových hodnot z SD karty.



V další části smyčky se provádí následující kroky:

1. Každou sekundu se získají hodnoty vnitřních hodin ESP32, které jsou synchronizovány z časového serveru NTP anebo pokud není dostupný z přesného zdroje reálného času DS3231.

Tyto údaje se ukládají do datové struktury. Zároveň se do ní uloží také informace, že došlo ke změně času.

2. Každou 10. sekundu se aktualizují data z teplotních čidel. První dvě slouží k trvalému měření vody v akváriu a z jejich údajů se vypočítává průměrná hodnota pro porovnání s hodnotou nastavené teploty vody, která je v hodinových intervalech pravidelně načítána z SD karty ze souboru setOTO.txt. Pokud dojde k odpojení nebo ke ztrátě komunikace s čidly 1 a 2, postup je následující:

Jestliže je platná hodnota teploty alespoň z jednoho čidla, bere se tento údaj jako průměrná teplota určená pro komparaci s teplotou nastavenou. Pokud jsou data z obou čidel neplatná, jako průměrná teplota se použije teplota nastavená, čímž se zamezí hrozbě přehřátí vody v akváriu. Třetí čidlo lze libovolně použít pro měření teploty podle potřeb akvaristy. Hodnoty těchto tří čidel jsou rovněž ukládána do datové struktury.

Průměrná teplota a teplota nastavená se uplatňuje při udržování stabilní teploty v akváriu. A to následujícím způsobem. V pravidelných časových intervalech je kontrolován rozdíl mezi nimi a podle výsledku je zapínán přes digitální výstup ESP32 číslo 25 a jím ovládané relé, akvaristické topení. Stav tohoto výstupu je zároveň indikován na stavovém displeji a na webovém rozhraní. Toto automatické topení je možno deaktivovat, buďto z webového rozhraní, anebo v souboru setOTO.txt.

3. Nastavení osvětlení. Je provedeno pouze pokud jsou rozdílné hodnoty mezi polem skutečných hodnot a polem nastavených hodnot. Když ano, je zkontrolováno, zda uplynul časový interval mezi jednotlivými kroky. V jednoduchosti to probíhá takto:

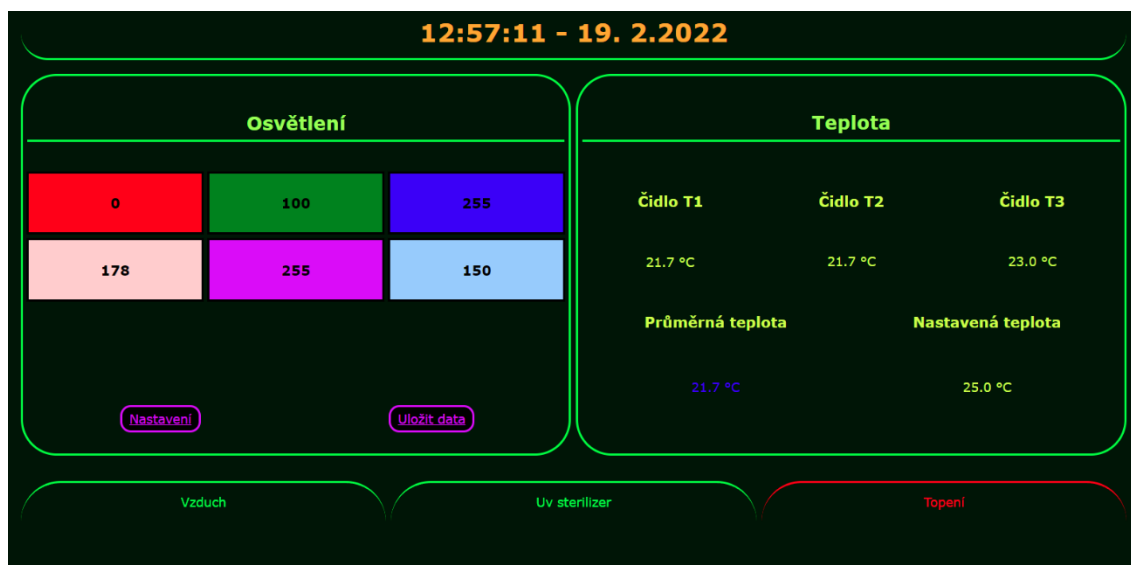
Hodnoty osvětlení jsou stejné, stav osvětlení se nemění. Pokud však jsou rozdílné, dojde k porovnání jednotlivých hodnot mezi oběma poli. Pokud je hodnota prvku se stejným pořadovým číslem v obou polích stejná, osvětlení zůstává na stávajícím stavu. Pokud ale hodnota prvku v poli nastavení je větší než hodnota prvku v poli současném, hodnota prvku v poli současném se zvedne o 1. Pokud je to obráceně, hodnota prvku se o 1 sníží. Toto se opakuje u všech šesti prvků v poli. Když jsou hodnoty v obou polích dále rozdílné, k dalšímu cyklu dojde po vypršení intervalu mezi kroky. Toto se opakuje, dokud se hodnoty v obou polích nerovnaj. Interval mezi jednotlivými kroky změny osvětlení je nastaven na 10 s, pokud ke změně v poli nastavených hodnot došlo při aktualizaci nastavení osvětlení ze souboru SetOTO.txt. Pokud ke změně dojde z webového rozhraní, interval mezi kroky se sníží na 1 s.

4. Pokud došlo ke změně času a datumu, teploty vody, hodnoty osvětlení anebo stavu výstupu, jsou tyto informace odeslány do webového rozhraní pomocí události odeslané serverem a slouží pro aktualizaci zobrazených hodnot.

5. Každých 10 s se ukládají hodnoty času, teploty, osvětlení a stavu výstupu na SD kartu do souboru saveData.csv, odkud je možné ho přes webové rozhraní uložit do počítače.

## 5.4 Webové rozhraní

Stránka HTML je odzkoušena v prohlížeči Firefox a Opera PC verze na Obr. 32 a navržena pro rozlišení 1920×1200. Do webového rozhraní je možný přístup zadáním IP adresy, která se po zapnutí jednotky OTO zobrazuje po dobu 10 s včetně názvu WIFI sítě a kanálu, na který je OTO připojen na stavovém displeji. Tyto informace je též možné zobrazit na již běžící jednotce, pomocí tlačítka. Po dobu jeho stisknutí se tyto informace zobrazí na displeji. Po uvolnění tlačítka dojde k návratu na původní obrazovku.



Obr. 32: Webové rozhraní jednotky OTO

Aktualizace stránky probíhá pomocí události odesílané serverem dle Obr. 33. Když má server nové hodnoty tak je použitím SSE odešle klientovi. HTML stránka obsahuje kód, který tuto událost zachytí a zpracuje. Příkaz „source.addEventListener('aktData', function(e) {console.log("aktData\_", e.data);“ zpracuje příchozí data, která jsou ve formátu JSON. Tyto údaje jsou rozděleny na jednotlivé hodnoty a jsou vloženy do patřičných míst na HTML stránce. Jedná se o hodnoty času a datumu, které jsou zobrazeny uprostřed v horní části stránky. Nalevo pod nimi jsou hodnoty osvětlení jednotlivých barev. Pokud se skutečné hodnoty liší od nastavených, jsou uvedeny v závorce vedle hodnot nastavených. Na pravé straně jsou hodnoty teploty čidel T1-T3, průměrná a nastavená teplota. Hodnoty jednotlivých výstupů ovlivňují barvy odpovídajících tlačítek. Při zapnutém vstupu je barva odpovídajícího tlačítka a textu červená. V opačném případě zelená. Výjimku tvoří tlačítko Topení, kde červená barva rámečku značí zapnutí automatické kontroly teploty vody v akváriu a zelená jeho vypnutí. O tom, zda je fyzický výstup topení zapnut nebo vypnut, informuje barva jeho textu.



```

274 // Aktualizace hodnot osvětlení a teploty pomocí serveru.
275 if (!!window.EventSource) {
276     var source = new EventSource('/events_OTO');
277
278     source.addEventListener('open', function(e) {console.log("Events Connected"); }, false);
279     source.addEventListener('error', function(e) {if(e.target.readyState != EventSource.OPEN)
280         {console.log("Events Disconnected");}}, false
281     );
282     source.addEventListener('message', function(e) {console.log("message", e.data); }, false);
283     // Vypíše přijatá data do konzole a následně je zpracuje.
284     source.addEventListener('aktData', function(e) {console.log("aktData_", e.data);
285

```

Obr. 33: Kód pro aktualizaci hodnot osvětlení a teploty

Kromě toho, že web stránka přijímá a zobrazuje údaje přijaté ze serveru, umožňuje také uživateli většinu těchto hodnot měnit. Tyto uživatelské změny jsou však platné pouze do konce aktuální hodiny. Mají-li být tyto změny trvalé, je nutné editovat řídicí soubor což je popsáno v kapitole Webová stránka pro nastavení hodnot.

Mezi parametry, které jdou měnit, patří samozřejmě osvětlení. Postup změny intenzity barvy je následující:

- Kliknutím na tlačítko konkrétní barvy se změní jeho orámování z černé na žlutou a opačně. Je-li barva rámečku žlutá, posuvník umístěný pod sekci osvětlení se nastaví na hodnotu aktuálně vybrané barvy. Jeho posouváním je možné měnit hodnotu vybrané barvy od 0 do 255. Změna hodnoty vybrané barvy se okamžitě projeví v textové části zvoleného tlačítka. Je-li aktivovaných více barevných kanálů, změnou hodnoty posuvníku se mění všechny hodnoty u všech aktuálně označených. Pro zafixování hodnoty daného kanálu je potřeba kliknutím změnit jeho rámeček na černý.
- V sekci Osvětlení se pod prvky pro ovládání barev nachází ještě dva HTML odkazy ve formě fialových tlačítek pojmenované Nastavení a Uložit data. Použitím těchto odkazů se stránky přesměrují na nastavení OTO anebo na stažení datových souborů.
- V pravé části HTML stránky se zobrazují hodnoty třech teplotních čidel, nastavená hodnota topení a průměrná hodnota vypočítaná z čidel T1 a T2 umístěnými v akváriu. Tato hodnota mění barvu, čímž uživatele spolu s tlačítkem topení vizuálně upozorňuje na stav teploty vody. Je-li barva modrá, je teplota vody nižší než nastavená, a naopak je-li teplota vody vyšší, je barva červená. Žlutá signalizuje, že teploty jsou vyrovnané.
- Ve spodní části HTML stránky jsou 3 tlačítka pro ovládání výstupu (T-U-V). Jejich barva signalizuje, zda je daný výstup zapnutý nebo vypnutý (červená značí vypnuto, zelená zapnuto).

Při aktualizaci údajů uživatelem jsou tyto hodnoty odeslány ke zpracování na web server.

#### 5.4.1 Webová stránka pro nastavení hodnot

Při volbě tlačítka Nastavení, budeme přesměrováni na web stránku s názvem: „Úprava řídicího souboru“ na Obr. 35 a na stavovém displeji se zobrazí nápis „Cekejte prosím, nastavuje se OTO“ viz Obr. 34.



Obr. 34: LCD displej se zprávou při zobrazení webového rozhraní řídicí jednotky OTO (úpravy hodnot)

Úprava řídicího souboru:								
Zobrazit	< nastavení ->		Odeslat	Posloupnost			Zpět	
Hodina	Hodnota osvětlení						Nastavené teplota	Výstupy V-U-T
	Červená	Zelená	Modrá	Teple bílá	Ultra fialová	Studeně bílá		
00	000	000	000	000	000	000	022	111
01	000	000	000	000	000	000	022	111
02	000	000	000	000	000	000	023	111
03	000	000	000	000	000	000	023	111
04	000	000	000	000	000	000	023	111
05	064	000	000	000	000	000	023	100
06	128	000	036	051	036	000	024	001
07	191	025	073	102	073	000	024	001
08	255	050	109	153	109	000	024	001
09	191	074	146	204	146	150	024	001
10	128	099	182	255	182	150	025	001
11	064	100	219	216	219	150	025	001
12	000	100	255	178	255	150	025	001
13	043	100	213	139	213	150	025	001
14	085	100	170	100	170	150	024	100
15	128	100	128	086	128	150	024	001
16	170	080	085	071	085	113	024	001
17	213	060	043	057	043	075	024	001
18	255	040	000	043	000	038	023	001
19	170	020	000	029	000	000	023	001
20	085	000	000	014	000	000	023	001
21	000	000	000	000	000	000	023	101
22	000	000	000	000	000	000	022	101
23	000	000	000	000	000	000	022	111

Obr. 35: Webové rozhraní řídicí jednotky OTO (úpravy hodnot)

Zde se nám otevře tabulka, která obsahuje data pro nastavení OTO (pokud se tak výjimečně nestane, lze použít tlačítko Zobrazit pro opětovné načtení dat).

V levém sloupci jsou jednotlivé hodiny dne od nuly do dvaceti tří, dále tabulka obsahuje šest sloupců s nastavením jednotlivých barev červené, zelené, modré, teple bílé, ultra fialové a studeně bílé, následují hodnoty pro nastavení teploty a v dalším sloupci se nastavují stavy jednotlivých výstupů (T-U-V). Hodnoty barevných kanálů je možné nastavovat v rozmezí 0-255, nastavení teploty může mít hodnotu od dvaceti do třiceti stupňů celsia a výstupy T-U-V, kde T je aktivace nastavené teploty, mohou nabývat hodnot 1 nebo 0, U je sterilizér, který UV zářením zbavuje vodu choroboplodných zárodků a V je vzduchování neboli okysličování vody.

Tyto hodnoty se dají měnit buď po jednotlivých buňkách, když myši vybereme příslušnou hodnotu, tak se ve spodní části tabulky zobrazí nový řádek s označenou hodnotou, kterou můžeme v daném rozmezí změnit. Toto rozmezí je kontrolováno a není možné uložit hodnotu mimo povolený rozsah. Na případné odchylky nás web stránka upozorní informačním oknem.

Hodnoty je také povoleno měnit po jednotlivých sloupcích, kdy myši se označí hlavička sloupce a poté do nového řádku, který se objeví na konci tabulky, se zapíše nová hodnota.

Změna hodnot ve sloupcích v kombinaci řádků se provede výběrem sloupce jako v předchozím případě, ale navíc i s označeným řádkem.

Na konci tabulky se vytvoří nový řádek se stejným číslem (hodin), který obsahuje hodnoty jednotlivých vybraných sloupců na zvoleném řádku. Při vložení údaje do zvýrazněné buňky, se tato hodnota zapíše do vybraného řádku a řádků nad ním až do dalšího vybraného řádku nebo začátku zvoleného sloupce. Pokud je vybraných řádků více, stejná hodnota se uloží od prvního označeného k dalšímu vybranému směrem nahoru. Těchto řádků je možné vybrat maximálně 6. Při překročení počtu řádků vyskočí upozornění informačním oknem.

Pro změnu výběru sloupců, je potřeba odstranit všechny řádky ve spodní části tabulky. To se provede opětovnou volbou jednotlivých již vybraných řádků. Poté je umožněno přidání nebo odebrání sloupců.

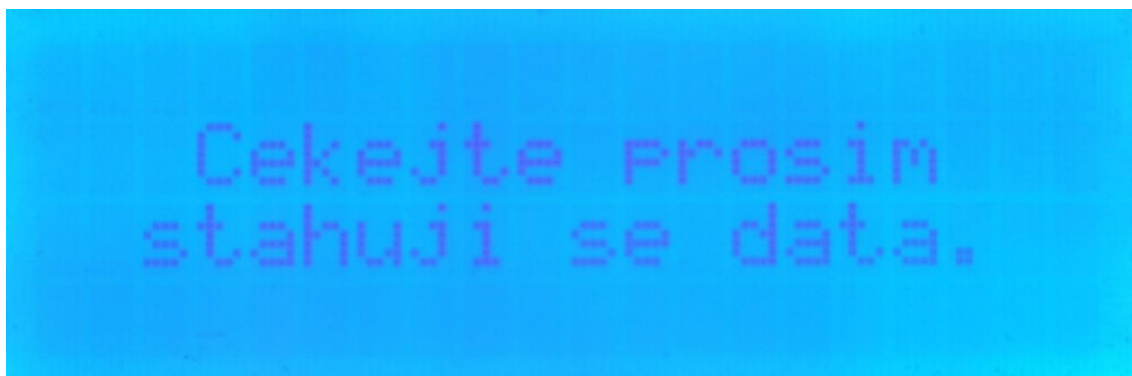
Při výběru řádků a sloupců je možné použít tlačítko posloupnost. To způsobí, že hodnoty nebudou mezi jednotlivými označenými řádky stejné, ale provede se výpočet lineární posloupnosti a jednotlivé hodnoty budou dosazeny do aktuálního sloupce tabulky.

Pokud je uživatel s celkovým nastavením spokojen, odešle aktualizované hodnoty do systému OTO pomocí tlačítka Odeslat. Tímto dojde k aktualizaci systému a zobrazí se úvodní web stránka.

Jestliže se z nějakého důvodu uživatel rozhodne, že aktualizaci neprovede, k návratu na hlavní stránku použije tlačítko Zpět, čímž dojde ke ztrátě změn provedených v tabulce.

#### 5.4.2 Webová stránka pro ukládání dat

Na tuto web stránku je možné se dostat z hlavního panelu použitím tlačítka Uložit data. Přechodem na tuto stránku se na stavovém displeji se zobrazí nápis „Cekejte prosim stahuji se data.“ Obr. 36 a na hlavní stanici vytvoří soubor, který obsahuje seznam všech dostupných souborů s uloženými daty na SD kartě dle Obr. 37.



Obr. 36: LCD displej se zprávou při zobrazení webového rozhraní řídicí jednotky OTO (stahování souboru)

Získat z OTO seznam souborů s daty
Zpět

error  
progress  
výsledek

Smazat vybraný soubor z OTO

**Datové soubory uložené na SD kartě.**

*Měsíc: 09*

20210911	20210915	20210917	20210918	20210919	20210920
20210922	20210923	20210924	20210925	20210926	20210927
20210928					

*Měsíc: 01*

20220106	20220108	20220109	20220110	20220111	20220112
20220113	20220116	20220118	20220120	20220121	20220122
20220125					

*Měsíc: 02*

20220216	20220217	20220218			
----------	----------	----------	--	--	--

Obr. 37: Webové rozhraní řídicí jednotky OTO (stahování souboru)

Tato činnost (podle počtu souborů) může trvat delší dobu. Proto v tomto čase se na této stránce zobrazuje animace s nápisem: Čekejte prosím, vytváří se seznam souborů. Když je soubor vytvořen a jeho obsah je načten do proměnné na stránce, tato animace zmizí a objeví se následující:

- Zobrazí se tlačítko pro opětovné načtení souboru, výstupní pole s názvem error, kde se později při ukládání souboru objevují případné chyby v komunikaci.
- Další vstupní pole s názvem progress zobrazuje při ukládání vybraného souboru grafický průběh, % stažení, rychlost stahování a celkovou dobu přenosu. Po skončení přenosu se ve 3. výstupním poli zobrazí počet přenesených bytů.
- Dále se zobrazí tlačítko pro volbu, zda stahovaný soubor zůstane dále na SD kartě anebo bude vymazán.
- Ve zbylé části HTML stránky se vytvoří tabulka, která obsahuje názvy jednotlivých datových souborů. Tyto jsou seřazeny podle datu pořízení a rozmístěny pro lepší přehlednost do jednotlivých měsíců.

Pro stažení souboru do PC jako kopii je potřeba vybrat název souboru s příslušným datem a kliknutím na něj započne stahování. Po stažení souboru jeho zobrazení v tabulce zešedne, což indikuje, že daný soubor byl již uložen.

Další možností je přesunutí souboru do PC zvolením tlačítka Smazat vybraný soubor z OTO a následně bude zvolen konkrétní požadovaný soubor. Po přesunutí souboru, jeho zobrazení v tabulce taky zešedne ale navíc se přes něj vytvoří červený křížek, což uživatele informuje o tom, že soubor již není na stanici OTO, a tudíž není možné ho opětovně stáhnout.

Pokud již není požadováno ukládání dalších souborů, tlačítkem Zpět umístěném v pravém horním rohu bude uživatel vrácen na hlavní stránku.

## 6 Ověření funkčnosti

Bylo provedeno na akváriu o objemu cca 58 l vody. Z důvodů zkušebního provozu a předpokladu častého zásahu do nastavení řídicí jednotky OTO nebylo osazeno faunou ani flórou.

Po propojení jednotlivých celků, tj. připojení panelu s osvětlením, topení, vzduchovacího kompresoru a UV sterilizeru (místo něho bylo zapojena lampička), bylo přistoupeno k samotnému otestování jednotlivých okruhů.

Po nahrání adresářové struktury a jednotlivých souborů na SD kartu byl vložen do souboru setup.txt do řádků „ssid: XXXXXX: a password: XXXXXX: “ název a heslo WIFI sítě.

Po zapnutí zdroje byla po automatické kontrole na displeji řídicí jednotky zobrazena IP adresa přidělená DHCP serverem. Po jejím zadání do prohlížeče na PC byla zobrazena úvodní stránka, na které se měnil čas, zobrazila se teplota ze 3 čidel a počáteční nastavení LED osvětlení. Kliknutím myši na jednotlivé výstupy v dolní části web stránky bylo otestováno, zda se mění jejich hodnoty mezi zapnuto a vypnuto a zda těmto stavům odpovídají i stavy fyzických výstupů. Po této kontrole, která byla úspěšná, byly přes webové rozhraní zadány hodnoty osvětlení, požadované teploty a nastavení výstupů pro jednotlivé hodiny dne. Toto nastavení bylo odesláno do řídicí jednotky. Po vypnutí jednotky byla vyjmuta SD karta a následně soubor s názvem setOTO.txt, nacházející se na SD kartě, byl zobrazen na PC. Hodnoty v souboru odpovídaly hodnotám v tabulce, která byla vytvořena na webovém rozhraní viz Obr. 38.

```
00:000:000:000:000:000:022:007
01:000:000:000:000:000:022:007
02:000:000:000:000:000:022:007
03:000:000:000:000:000:022:007
04:000:000:000:000:000:023:005
05:100:000:000:000:010:000:023:005
06:255:000:036:000:045:000:024:005
07:213:005:073:000:080:005:024:005
08:170:010:109:100:115:047:024:001
09:128:000:146:088:150:088:025:001
10:085:000:182:075:185:130:025:001
11:043:000:219:063:220:172:025:001
12:000:000:255:050:255:213:025:001
13:036:000:213:059:224:255:025:001
14:073:000:170:067:194:170:025:001
15:109:000:128:076:163:085:025:001
16:145:000:085:084:133:000:025:001
17:181:010:043:093:102:000:024:001
18:218:010:000:101:071:000:024:005
19:254:000:000:110:041:000:023:005
20:100:000:000:000:010:000:023:007
21:000:000:000:000:000:000:022:007
22:000:000:000:000:000:000:022:007
23:000:000:000:000:000:000:022:007
```

Obr. 38: Hodnoty pro nastavení v souboru setOTO.txt

Tímto krokem byla ověřena funkčnost přenosu dat mezi jednotkou OTO a PC. Po návratu na hlavní stránku byly otestovány jednotlivé kanály osvětlení a to tak, že manuálně byly měněny úrovně osvětlení jednoho kanálu po druhém a zároveň sledováno, zda dochází ke změně intenzity osvětlení na osvětlovacím tělese a také zda se mění správný kanál (led páska). Po této kontrole bylo zařízení resetováno a necháno pracovat v automatickém režimu, přičemž během dne byly sledovány následující parametry. V různých časových intervalech proběhla kontrola teploty vody porovnáním průměrné teploty, kterou ukazovala čidla T1 a T2 s laboratorním teploměrem v Tab. 9.

Tab. 9: Hodnoty naměřené v akváriu

Čas [-]	Laboratorní teploměr [°C]	Průměr dvou čidel [°C]	Rozdíl teplot [°C]
10:20	25,5	25,2	0,3
10:50	24,9	25,1	-0,2
11:20	25	24,7	0,3
11:50	25	24,6	0,4
12:20	24,9	24,6	0,3
12:50	25,2	24,9	0,3
13:20	25,4	25,1	0,3
13:50	25,5	25,1	0,4
14:20	25,1	24,8	0,3
14:50	25	24,7	0,3

Dále bylo kontrolováno, zda se jednotlivá čidla mezi sebou neliší o více než 1 °C, neboť podle dokumentace by v intervalu teplot od -10 °C do + 85 °C měla být chyba měření čidla maximálně  $\pm 0,5$  °C. Čidla T1 a T2 tuto podmínku splňovala ale T3 měřilo s větší chybou. Průměrná teplota a teplota změřená laboratorním teploměrem se v rozsahu měřených teplot lišila maximálně o  $\pm 0,4$  °C. Laboratorní teploměr, který byl použit, má udávanou přesnost měření  $\pm 1$  °C.

Také bylo sledováno, zda se automaticky vždy v celou hodinu mění intenzita osvětlení a zda rozsvícené led pásy odpovídají barevným kombinacím v řídicím souboru na Obr. 38 a též zda výstupy T-U-V jsou v souladu. Dále bylo kontrolováno, zda hodnoty osvětlení jednotlivých kanálů, teplot a výstupů jsou stejné jak v řídicím souboru, tak na webovém rozhraní a také na stavovém LCD displeji na řídicí jednotce OTO. Takto byla jednotka v činnosti několik dnů ke získání souboru dat k dalšímu zpracování. Následující hodiny a dny bylo zařízení kontrolováno pouze namátkově. Poté byl přes webové rozhraní stažen jeden kompletní soubor s daty a z těchto dat byly vytvořeny grafy na Obr. 39 a Obr. 40.

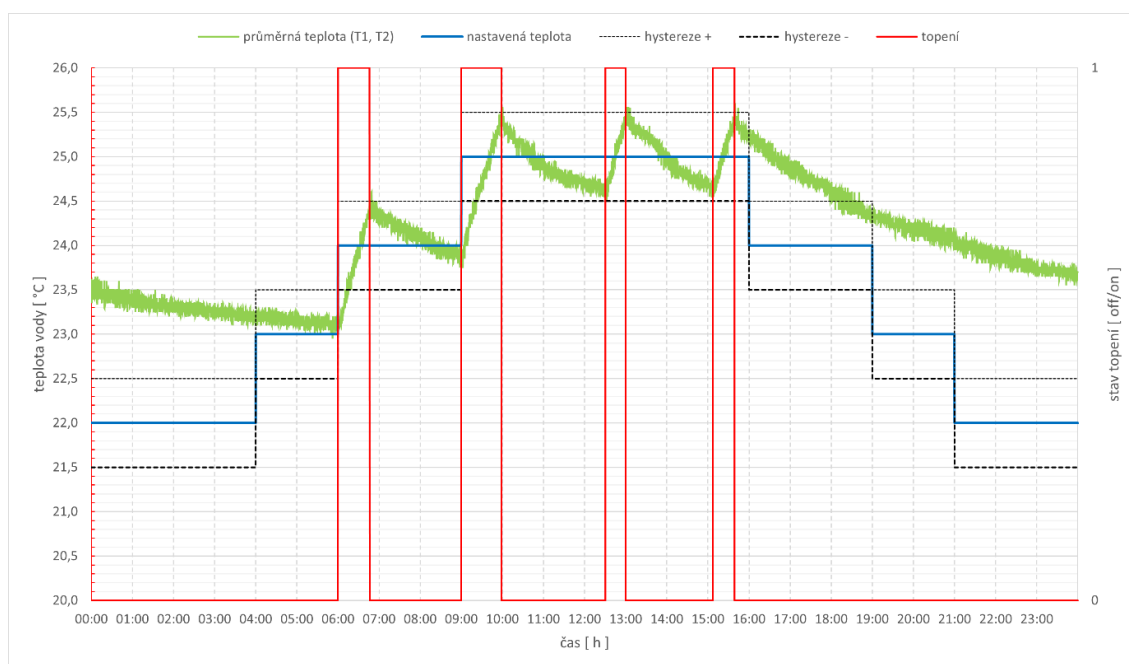
Na grafu Obr. 39 je vidět, jak se měnila teplota vody v akváriu během dne. Zeleně je vyznačen průběh průměrné teploty změřené jednotkou. Modrá čára ukazuje nastavení teploty v průběhu dne podle řídicího souboru a červeně je znázorněno spínání topného tělesa. Čerchovaná černá čára je hystereze, která zabraňuje častému spínání topného tělesa a je nastavena na  $\pm 0,5$  °C.



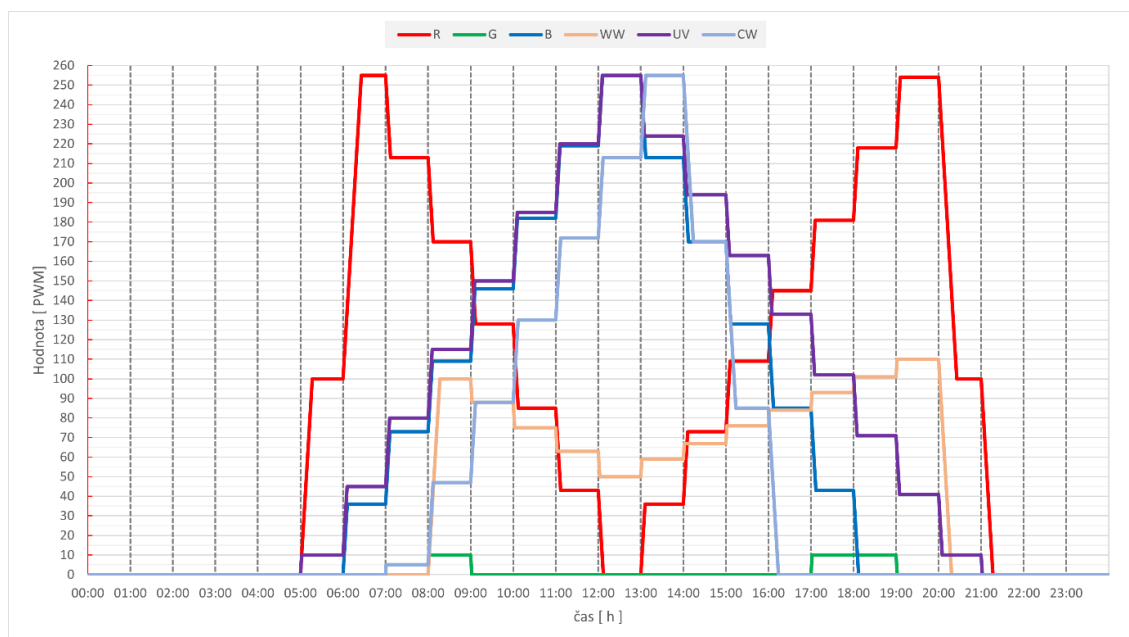
Kromě ní ovlivňují teplotu vody a tím i výsledný graf následující veličiny:

- celkové množství vody v akváriu
- výkon topného tělesa
- cirkulace vody v akváriu
- teplota okolního prostředí, v němž se akvárium nachází.

Ve druhém grafu Obr. 40 jsou zobrazeny průběhy signálů PWM pro jednotlivé kanály led osvětlení. Jejich změnou lze nasimulovat různé barevné kombinace světla a experimentálně si ověřit, při jaké intenzitě a barvě světla se bude nejlépe dařit vodním rostlinám, neboť každá rostlina má jiné nároky na osvětlení.



Obr. 39: Graf průběhu teploty vody v akváriu během dne

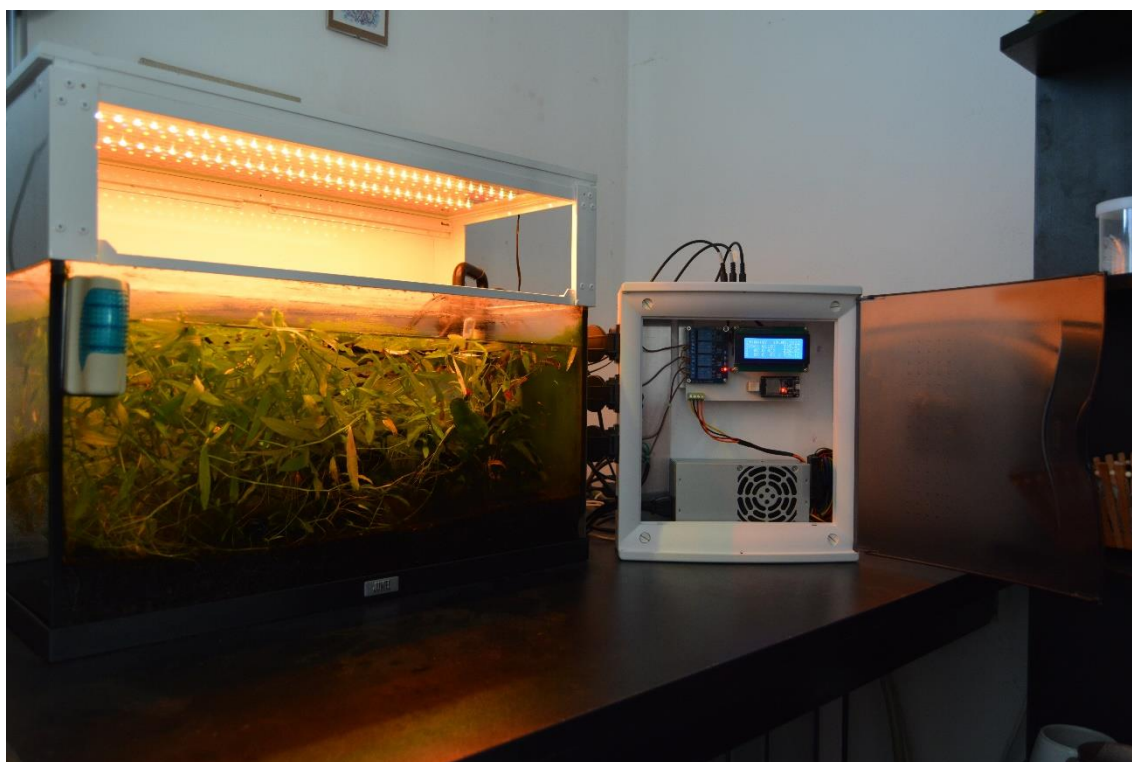


Obr. 40: Graf průběhu jednotlivých kanálů osvětlení akvária během dne

## 7 Závěr

Cílem této práce byl návrh a realizace řídicí jednotky vnitřního prostředí akvária, která by vyhovovala velké skupině akvaristů a byla poměrně snadno ovladatelná a nastavitelná. Zvolené řešení je schopno nejen splnit dané požadavky jako je zapínat a vypínat osvětlení a teplotu, ale i měnit tyto hodnoty dle uživatelského nastavení. Osvětlení umožňuje ovládat 6 nezávislých kanálů pro jednotlivé barvy v rozsahu PWM intenzity od 0-255, měřit teplotu třemi čidly, kde 2 jsou umístěny v akváriu a měří průměrnou teplotu vody a 3. slouží pro umístění do konkrétního bodu pro měření jeho teploty. Získané hodnoty uložené na SD kartě i nastavení řídicí jednotky jsou přístupné přes webové rozhraní.

Finální výrobek byl realizován na univerzálním PCB a vestavěn do rozvodné krabice typu 3824 Tapa Opaca. Pohled na celkové řešení je na Obr. 41. Detail řídicí jednotky na Obr. 42.



Obr. 41: Celkový pohled na řídicí jednotku OTO s akváriem





Obr. 42: Detailní pohled na finální verzi řídicí jednotky

Možné vylepšení by bylo v implementaci tabulek pro kalibraci teplotních čidel, podle kterých by se upravovala naměřená hodnota.

Použitím peristaltických čerpadel lze automaticky podle nastavení dodávat do akvária živiny pro rostliny. Tím by se v delším časovém intervalu dalo z uložených dat vysledovat závislosti růstu rostlin na typu osvětlení, teplotě a množství živin. Podle získaných dat by se jednotka OTO dala nastavit na míru různých prostředí.

Přidáním modulu na měření PH vody by se mohl ovládat elektromagnetický ventil, který řídí přidávání CO<sub>2</sub> z tlakové lahve do vody za účelem zlepšení výživy rostlin (využití hlavně u akvárií s převahou rostlin).

Další moduly, o které by se mohla rozšířit funkčnost jednotky OTO:

- měření kyslíku ve vodě
- vodivost vody
- oxidačně redukční potenciál

Tyto moduly komunikují po sběrnici I2C a nebylo by těžké je zakomponovat do řídicí jednotky.

## 8 Seznam použitých zdrojů

- [1] Aquat Atlantis Luminus dvoukanálový controller. In: *Sklorex* [online]. Koperníková 19 615 00 Brno - Židenice: SkloREX Akvárium, spol. s r.o., 1992 [cit. 2022-03-31]. Dostupné z: <https://www.sklorex-akvarium.cz/p7094-easyled-luminus-controller-stmivac-aquatatlantis.html>
- [2] Aquat Atlantis EasyLed Control 2 Plus stmívač. In: *Sklorex* [online]. Koperníková 19 615 00 Brno - Židenice: SkloREX Akvárium, spol. s r.o., 1992 [cit. 2022-03-31]. Dostupné z: <https://www.sklorex-akvarium.cz/p5977-aquatatlantis-easyled-control-2-plus-stmivac.html>
- [3] Programovatelný stmívač pro LED osvětlení s WiFi ovládáním. In: *INVITAL Rostlinna-akvaria* [online]. Opavská 397, Hradec nad Moravicí 747 41: INVITAL Aqua s.r.o., 2000 [cit. 2022-03-31]. Dostupné z: <https://www.rostlinna-akvaria.cz/eshop/led-osvetleni-akvaria/programovatelny-stmivac-pro-led-osvetleni-s-wifi-ovladanim>
- [4] *ESP32 Series: Datasheet* [online]. Shanghai: Espressif Systems, 2022 [cit. 2022-03-31]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)
- [5] *ESP 32-WROOM-32: Datasheet* [online]. Shanghai: Espressif Systems, 2018 [cit. 2022-05-19]. Dostupné z: [https://cdn.sparkfun.com/assets/learn\\_tutorials/8/0/4/esp32-wroom-32\\_datasheet\\_en.pdf](https://cdn.sparkfun.com/assets/learn_tutorials/8/0/4/esp32-wroom-32_datasheet_en.pdf)
- [6] Getting Started with the ESP32 Development Board. In: *Randomnerdtutorials* [online]. Porto: Rui Santos and Sara Santos, 2013 [cit. 2022-05-20]. Dostupné z: <https://randomnerdtutorials.com/getting-started-with-esp32/>
- [7] *DS3231: Extremely Accurate I2C-Integrated RTC/TCXO/Crystal* [online]. San Jose, CA 95134 USA: Maxim Integrated Products, Inc., 2015 [cit. 2022-03-31]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>
- [8] *DS18B20: Programmable Resolution 1-Wire Digital Thermometer* [online]. San Jose, CA 95134 USA: Maxim Integrated Products, Inc, 2019 [cit. 2022-03-31]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [9] 1-Wire. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2022-03-31]. Dostupné z: <https://cs.wikipedia.org/wiki/1-Wire>
- [10] *WS2811: Signal line 256 Gray level 3 channel Constant current LED drive IC* [online]. Čína: WORLDSEMI, 2010 [cit. 2022-03-31]. Dostupné z: <https://cdn-shop.adafruit.com/datasheets/WS2811.pdf>
- [11] LCD displej-blokový diagram. In: *Importedelectronics* [online]. [cit. 2022-05-16]. Dostupné z: <https://www.importedelectronics.com/products/lcd-2004-yellow-serial-spi-20x4-lcd2004-character-module-display-screen-arduino#images-13>
- [12] *PCF8574; PCF8574A: Remote 8-bit I/O expander for I2C-bus with interrupt* [online]. 2013 [cit. 2022-05-20]. Dostupné z: [https://www.nxp.com/docs/en/datasheet/PCF8574\\_PCF8574A.pdf](https://www.nxp.com/docs/en/datasheet/PCF8574_PCF8574A.pdf)

## Seznam použitých zkratek

OTO	Ovládání Teploty a Osvětlení
T-U-V	Výstupy Topení, Uv sterilizer, Vzduchování
MCU	MicroController Unit - Jednočipový počítač
CPU	Central Processing Unit-Centrální procesorová jednotka
RAM	Random Access Memory - Paměť s přímým přístupem
EEPROM	Electrically Erasable Programmable Read-Only Memory - Elektricky vymazatelná paměť pouze pro čtení
SRAM	Static Random Access Memory - Statická paměť
ROM	Read-Only Memory - Paměť pouze pro čtení
IrDA	Infrared Data Association - Komunikační infračervený port
I2C	Inter-Integrated Circuit - Počítačová sériová sběrnice
SPI	Serial Peripheral Interface - Sériové periferní rozhraní
DMA	Direct Memory Access - Přímý přístup do paměti
Mbps	Mega bit per second - Mega bit za sekundu
dBm	decibel-milliwatts - Jednotka absolutní úrovně vztažené k výkonu 1 mW
PCB	Printed Circuit Board - Deska plošných spojů
SD	Secure Digital - Paměťová karta
NTP	Network Time Protocol - Protokol pro synchronizaci vnitřních hodin počítačů
LCD	Liquid Crystal Display - Displej z tekutých krystalů
UV	Ultra violet - Ultra fialový
LED	Light-Emitting Diode - Elektroluminiscenční dioda
RTC	Real-time clock - Hodiny reálného času
GPIO	General-Purpose Input/Output - Univerzální vstupní/výstupní pin
SoC	System on a chip - Počítač integrovaný v jediném integrovaném obvodu
AC	Alternating current - Střídavý proud
DC	Direct current - Stejnosměrný proud
WEP	Wired Equivalent Privacy
WPA2	Wireless Protected Access
IPAdr	IP address - Adresa zařízení v síti
IP	Internet Protocol - Internetový protokol
HTTP	Hypertext Transfer Protocol - Internetový protokol
SSE	Send Server Event - Události odeslané serverem
JSON	JavaScript Object Notation - JavaScriptový objektový zápis

## **Seznam příloh**

- OTO