Міністерство освіти і науки України

Управління освіти і науки Рівненської обласної державної адміністрації

Рівненська Мала академія наук учнівської молоді

Відділення комп'ютерних наук

Секція “Комп’ютерні системи та мережі”

**РОЗУМНИЙ АКВАРІУМ**

Роботу виконав:

учень 11 класу

Рівненської загальноосвітньої

школи І-ІІІ ступенів №1 імені Володимира Короленка

Чернюк Олександр Олександрович

Наукові керівники:

Ольховик Вадим Леонідович,

керівник гуртків профільної ІТ школи

Рівненського міського

Палацу дітей та молоді

Жуковський Віктор Володимирович,

к.т.н., доцент кафедри комп’ютерних наук та прикладної математики

Національного університету водного господарства та природокористування

Рівне 2021

# АНОТАЦІЯ



Чернюк Олександр Олександрович,

учень 11 класу Рівненської загальноосвітньої школи І-ІІІ ступенів № 1, учень профільної ІТ школи комунального закладу «Рівненський міський Палац дітей та молоді»

Рівненська Мала академія наук учнівської молоді, м. Рівне

Наукові керівники:

Ольховик Вадим Леонідович

керівник гуртків профільної ІТ школи комунального закладу «Рівненський міський Палац дітей та молоді»,

Жуковський Віктор Володимирович,

к.т.н., доцент кафедри комп’ютерних наук та прикладної математики

Національного університету водного господарства та природокористування,

РОЗУМНИЙ АКВАРІУМ

Дослідницьку роботу присвячено темі ІоТ пристроїв, що здатні керувати параметрами середовища в акваріумі через мережу Інтернет. Це дозволяє цілодобово моніторити стан справ та керувати віддалено з будь-якої точки світу.

Досліджено саму концепцію смарт акваріума, зроблено огляд існуючих комерційних розумних пристроїв для годування рибок в акваріумі.

На основі проведеного аналізу існуючих зразків, було спроектовано власний пристрій з використанням відомих електронних компонентів, реалізовано його програмне наповнення та виготовлено робочий макет корпусу за допомогою 3D принтера.

Програмна частина створена на платформі Arduino та включає в себе веб-сервер з користувацьким інтерфейсом у вигляді веб сторінки.

Результатом роботи став пристрій, що здатен працювати автономно, отримувати команди від власника та приймати необхідні рішення по догляду за акваріумом самостійно на основі складеного графіку.

**Ключові слова:** Інтернет речей, веб-сервер, веб-інтерфейс, розумний акваріум.

**ЗМІСТ**

[АНОТАЦІЯ](#_heading=h.gjdgxs) 2

[ВСТУП](#_heading=h.30j0zll) 5

[РОЗДІЛ 1](#_heading=h.1fob9te). [**Що таке Смарт-акваріум і як він працює?**](#_heading=h.jp9esiq08ow7)6

[РОЗДІЛ 2](#_heading=h.3znysh7). [**Подібні проєкти**](#_heading=h.ym8cv2a4ghw6)8

[2.1. Xiaomi Fish Tank Pro](#_heading=h.2et92p0) 8

[2.2. Xiaomi YouPin](#_heading=h.tyjcwt) 9

[2.3. Aquael Shrimp Set Smart](#_heading=h.3dy6vkm) 10

[РОЗДІЛ 3](#_heading=h.1t3h5sf). [**Використані модулі та їх роль у проєкті**](#_heading=h.dqe2k01615fq)13

[3.1. Плата ESP32](#_heading=h.4d34og8) 13

[3.2. Датчик температури](#_heading=h.2s8eyo1) 15

[3.3. Годинник реального часу](#_heading=h.17dp8vu) 16

[3.4. Рідкокристалічний дисплей](#_heading=h.26in1rg) 17

[3.5. Сервопривід](#_heading=h.lnxbz9) 18

[3.6. Принципова електрична схема та робочий прототип](#_heading=h.35nkun2) 19

[3.7 Система відеоспостереження](#_heading=h.1ksv4uv) 20

[РОЗДІЛ 4](#_heading=h.44sinio). [**Програмний код**](#_heading=h.7utjy5olshjo)22

[4.1. Веб-сайт для керування](#_heading=h.2jxsxqh) 22

[4.2. Прошивка мікроконтролера](#_heading=h.z337ya) 23

[РОЗДІЛ 5](#_heading=h.wntd7oja16xt). [**Проектування 3D моделей елементів корпусу**](#_heading=h.gehzvrtzuxkm)32

[ВИСНОВОК](#_heading=h.1y810tw) 36

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ](#_heading=h.4i7ojhp) 37

# ВСТУП

У сучасному світі програмування застосовується для вирішення завдань широкого спектру. Нині багато передових компаній створюють власні додатки, які можуть контролювати предмети, що знаходяться у будинку, дистанційно, а іноді взагалі й без людського втручання. Можемо взяти для прикладу smart-house, який включає у себе різний асортимент електричних приладів вашого будинку. Поки повертаєтесь з роботи, Ви можете за допомогою смартфону вмикнути *електрочайник* та *робот-порохотяг.* До приїзду на Вас вже чекатиме готовий теплий чай та чиста, прибрана кімната. Таких приладів може бути безліч, починаючи від холодильників та пральних машинок й закінчуючи звичайною електрогодівничкою для тварин.

Дивлячись на це все, будь-який мешканець запитає:  “Чи є взагалі такі проєкти, як smart-house, але тільки для акваріуму?” Над цим проєктом йде робота близько двох років. Минулорічна ціль, що постала перед проєктантами, - створення штучної імітації сонячного світла за допомогою LED-стрічки. А ціль, яка стоїть нині, – зробити електрогодівничку для риб, створити сайт, на якому можна було б дистанційно спостерігати за водним середовищем, розробити 3D-макет станції та надрукувати його на 3D-принтері.

Перед тим, як розпочати роботу з проєктування, було розглянуто ринок подібних пристроїв, ціна яких нас, сучасних споживачів, не влаштовувала. Тому було ухвалено рішення - спроєктувати “Smart aquarium” власноруч.

# РОЗДІЛ 1

# Що таке Смарт-акваріум і як він працює?

Акваріум – одне із найвідоміших видів захоплень людей, що приваблює нас своєю флорою і фауною. Цікаво спостерігати за плаванням риб, ростом рослин, пристосуванням організмів до нового середовища. Також акваріум є непоганим елементом декору, яким можна прикрасити свою кімнату. Але, як і за кожною домашньою твариною, за акваріумним середовищем потрібен ретельний догляд. Проте догляд за акваріумом є досить-таки складною процедурою. Тому розробка спрощеного варіанту догляду за акваріумом і стала метою наукової роботи.

Проєкт «Смарт-акваріум» має у собі відразу декілька корисних фунцій: LED-підсвітку з імітацією сонячного світла, електрогодівничку, камеру відеоспостереження, Wi-Fi керування, датчик температури і не тільки.

Перевагою цього проєкту є те, що головна конструкція живиться від звичайної 5-вольтової зарядки телефону, яка наявна майже у кожному будинку. Проте існує одне застереження: сила струму в такій зарядці повинна бути не менше 2 Ампер. Наприклад, можна взяти звичайну зарядку від компанії Xiaomi. Саме вона і стала в нагоді під час розробки вказаного проєкту.

Імітація сонячного світла відбувається завдяки LED-стрічки, яка живиться від мережі 5 вольт і керується ШІМ-сигналом. Завдяки платі ESP32, котра має роздільну здатність АЦП до 12-біт, ми маємо 2^12 рівнів управління освітлення в акваріумі. Тобто, стрічка може мати цілих 4096 видів яскравості, які можна використовувати у зростаючому або ж спадаючому порядку. Для спрощення підрахунків я обмежив бітність АЦП до 10-ти біт. Тобто, зараз є 1024 рівнів яскравості, де 0 - це вимкнення стрічки, а 1023 - це її повна потужність.

Електрогодівничка стала однією з найважчих цілей у вказаному проєкті. Повне моделювання та друк всіх елементів відбувався впродовж декількох тижнів. Зверху цієї конструкції розташована мала спеціальна ємність для корму, який потрапляє в акваріум завдяки руху шнека. Ця деталь обертається завдяки встановленому серво-приводу.  Перевагою цієї годівнички є те, що корм подається кожного дня у певну задану годину, без втручання людини. Якщо виникне бажання знову погодувати риб, то це можна буде зробити завдяки кнопці, яка розміщена на вебінтерфейсі контролера.

Всі ці дії контролює одна плата ESP32,  де знаходиться Wi-Fi модуль, за допомогою якого ми і хостимо сайт. У свою чергу на сайті Ви можете переглянути стан освітлення в акваріумі, встановити точний час вмикання та вимикання освітлення в середовищі, увімкнути годівничку за допомогою однієї кнопки та  встановити  час “обіду”.

# РОЗДІЛ 2

# Подібні проєкти

Подібних проєктів на ринку не дуже багато. Існує невеликий попит на такі середовища, адже інколи ціна проєктів сягає 75-600 доларів США. Ціна залежить від об’єму самого акваріума. Лідерами серед продажу вказаних речей, як не дивно,  є компанія Xiaomі зі своїми **Fish Tank Pro** (див. «Рис. 2.1») і **YouPin** (див. «Рис. 2.2»)**,** компанія AqueEl з **SHRIMP SET SMART** (рис 2.3)**.** Хоча на ринку існують менш відомі виробники, які також наздоганяють лідерів розумних акваріумів, створюючи нові функції та можливості з хорошою якістю, гарним дизайном за прийнятну ціну. Це такі компанії, як **Hoison** «Рис 2.4», **Silhouette Tetra** «Рис 2.5»та багато інших..

## 2.1. Xiaomi Fish Tank Pro

Компания Xiaomi представила новий розумний акваріум Xiaomi Smart Fish Tank Pro, який підтримує додаток Xiaomi Mijia. Користувачі зможуть отримувати різну інформацію прямо на свій смартфон.

Новий Xiaomi Smart Fish Tank Pro має габарити 382 x 166 x 229 мм, корпус виконаний з акрилового скла і білого ABS-пластика. Скло має глянцевий ефект і забезпечує більшу прозорість в порівнянні зі звичайними акваріумами.

Користувачі можуть підключити акваріум до звичайної розетки або портативного акумулятора. Xiaomi Smart Fish Tank Pro оснащений подвійною системою фільтрації, здатної очистити всю воду всього за кілька хвилин. Він також отримав модуль Wi-Fi для зв'язку зі смартфоном, який дозволяє управляти акваріумом.

У новому акваріумі є автоматична годівниця, яку також можна контролювати через додаток зі смартфона. Також новинка отримала сучасну систему підсвічування, здатну висвітлити внутрішню частину акваріума (користувачам доступні 1600 відтінків кольорів) [1].



Рис 2.1. Вигляд Xiaomi Smart Fish Tank Pro

## 2.2. Xiaomi YouPin

Створений SOBO акваріум Xiaomi YouP має розміри 34 х 28,5 х 19 см, тому він може легко знайти місце на будь-якому столі. Без води він має вагу 3,7 кг і має бак, що вміщає не більше 15 літрів. Цей акваріум зручний для дрібних рибок: за заявою виробника, він підходить для 12 риб розміром 4 см, 4 штук - 6 см або 2 штук - 10 см.

Скло виготовлене з високоякісних матеріалів, має антиблікове і надпрозоре покриття, забезпечуючи максимальну внутрішню видимість. Крім того, відсутність рамок підкреслює мінімалістичний дизайн, який дає простір для головних героїв всередині. Даний акваріум продається прямо на Aliexpress за ціною близько 35 € [2].



Рис 2.2. Вигляд Xiaomi YouPin акваріума

## 2.3. Aquael Shrimp Set Smart

Акваріуми Aquael Shrimp Set Smart призначені для розведення креветок і невеликих риб, а також для вирощування водних рослин. Вони виготовлені і скла високої прозорості і склеєні спеціалізованим прозорим силіконом, який гарантує міцність і герметичність швів. Поліпропіленова підкладка, на яку встановлюється акваріум, згладжує нерівності поверхні і допомагає рівномірному розподілу ваги спорядженого комплексу.

Акваріуми Aquael Shrimp Set Smart обладнані покривним склом, яке запобігає самовільне покидання акваріума його мешканцями, а також знижує випаровування води. Особливістю оновленого Aquael Shrimp Set Smart є система кріплення покривного скла - воно ніби ширяє над акваріумом, що надає всьому комплексу особливу чарівність.

Плоска форма оновленого світильника Leddy Smart надає елегантність всьому акваріумному комплексу. А що випромінюється світло c колірною температурою 8000 K сприяє росту рослин завдяки правильно підібраному спектру, а також підкреслює красу натурального забарвлення акваріумних мешканців [3].

Чистота води забезпечується мініатюрним фільтром Pat-Mini з зовнішньої губкою, яка безпечна навіть для самих маленьких креветок.

Температурний режим в акваріумі підтримується за допомогою нагрівача Aquael Comfortzone Fix.



Рис 2.3. Вигляд акваріума Aquael Shrimp Set Smart

Крім того перейти за посиланнями можна за допомогою QR-коду

**Fish Tank Pro**

**YouPin**



**SHRIMP SET SMART**

**Hoison**



**Silhouette Tetra**

****

# РОЗДІЛ 3

# Використані модулі та їх роль у проєкті

Під час розробки цього проєкту використовувалися наступні модулі:

1. ESP-32cam (головний контролер);
2. DHT-11 (датчик температури );
3. DS1307RTC(датчик точного часу);
4. LCD1602(монітор виводу);
5. Сервопривід (MG 996R).

Бібліотеки, що були використані під час розробки проєкту:

1. ESP32Servo.h(бібліотека для керування двигуном постійного струму );
2. SPIFFS.h (бібліотека для Wi-Fi мережі);
3. SimpleDHT.h(бібліотека для отримання інформації з датчику температури);
4. LiquidCrystal\_I2C.h / Wire.h (бібліотека для монітору);
5. DS1307RTC.h(бібліотека для датчика точного часу).

## 3.1. Плата ESP32

Також хотілося б звернути увагу найголовнішу частину цього проєкту - контролер. Чому вибір припав саме на ESP32? Чому ми не обрали інший контролер?

Плата ESP32 - це один із найкращих контролерів для навчання та проєктування, адже він прошивається за допомогою середовища Arduino IDE, має непогану потужність. Якщо її порівнювати з платами Arduino, такими, як Uno, Nano, Leo, Wemos та багатьма іншими, то ESP32 буде лідером за тактовою частотою, за об’ємом пам’яті. Якщо вже говорити про потужність, то для вказаного проєкту непоганим варіантом була б плата Raspberry Pi, але її ціна у кілька разів перевищує ціну обраного контролера. Тому ESP3CAM («Рис. 3.1»), маючи невеликі розміри та оптимальну ціну, стає прекрасним вибором для заявленого проєкту [4, 5].

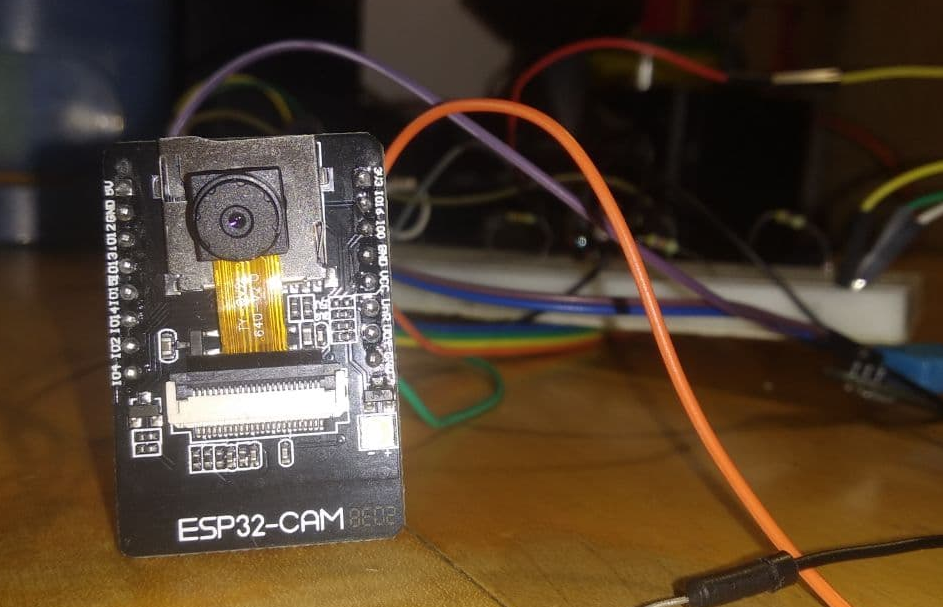


Рис 3.1. ESP32 в процесі монтажу

Технічні характеристики:

* WiFi 802.11b / g / n
* Bluetooth 4.2 LE з друкованою антеною, роз'єм u.FL. (IPX13)
* 32 Мбіт SPI флеш-пам'ять
* 4 Мб PSRAM
* Слот для мікро SD-карти до 4 Гб
* Габаритні розміри - 40.5 x 27 x 4.5 мм
* Вага - 10 гр.

Камера:

* роз'єм FPC
* Підтримка камер OV2640, OV7670
* Формат зображення - JPEG (підтримує тільки OV2640), BMP, відтінки сірого
* Світлодіодна підсвітка
* Розширення - UART, SPI, I2C, PWM
* кнопка скидання

Живлення - 5 вольт через контактний роз'єм [4].

## 3.2. Датчик температури

DS18B20 («Рис.3.2») - це герметичний датчик [6] з довжиною провідника 1 метр, використовується для вимірювання температури не лише в повітрі, а й під водою. Датчик DS18B20 володіє високою швидкодією, точністю, простотою, та герметичністю, що й дозволяє нам вимірювати температуру в акваріумі.



Рис 3.2. Датчик температури DS18B20

Технічні характеристики датчика DHT11

* Інтерфейс: One-Wire
* Живлення: DC 3,0 - 5,5 В
* Струм живлення: в режимі вимірювання 1mA.
* визначення вологості 20-80% з точністю 5%
* визначення температури від -55°С до +125°С (-67°F - +257°F) з точністю 0.5°
* частота опитування не більше 1.4 Гц (раз на 750 мсек.)
* довжина проводу: 1метр

## 3.3. Годинник реального часу

Нами було вибрано невеликий модуль DS1307 для підрахунку часу («Рис.3.3»). Він зібраний на базі мікросхеми DS1307ZN з реалізацією живлення від літієвої батарейки (LIR2032), що дозволяє працювати автономно протягом тривалого часу. Також на модулі, встановлена незалежна пам'ять EEPROM об'ємом 32 Кбайт (AT24C32). Мікросхема AT24C32 і DS1307ZN пов'язані загальної шиною інтерфейсом I2C [7].

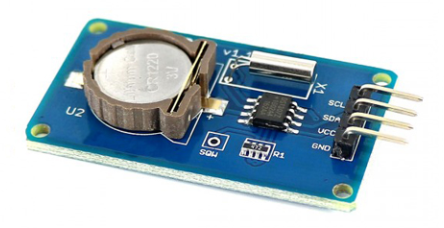


Рис 3.3. Годинник реального часу DS1307

Використання модуля DS1307 часто є дуже виправдано, наприклад, коли дані зчитуються рідко (як от в нашому випадку коли ми рідко кормимо рибок). Забезпечити безперебійне постійне живлення, наприклад плати ESP, на тривалий термін нераціонально, навіть при використанні батареї.

Завдяки своїй пам'яті і автономності, можна реєструвати події, (при автономному живленні) наприклад зміна температури і так далі, дані будуть зберігатися в пам'яті і їх можна буде зчитувати з пам'яті модуля. Так що модуль DS1307 часто використовують, коли контролерам Arduino/ESP/тощо необхідно знати точний час, для запуску якоїсь події і так далі. В нашому випадку це час корміння рибок і, відповідно, запуск шнекового механізму.

## 3.4. Рідкокристалічний дисплей

Рідкокристалічний дисплей (Liquid Crystal Display) скорочено LCD побудований на технології рідких кристалів. При проектуванні електронного пристроя, нам потрібно було вибрати недорогий пристрій для відображення інформації і також, важливим критерієм була наявність готових бібліотек для Arduino. З усіх доступних LCD дисплеїв на ринку, найбільш часто використовується LCD 1602A («Рис. 3.4»), який може відображати ASCII символу в 2 рядки (16 знаків в 1 рядку) кожен символ в вигляді матриці 5х7 пікселів [8].

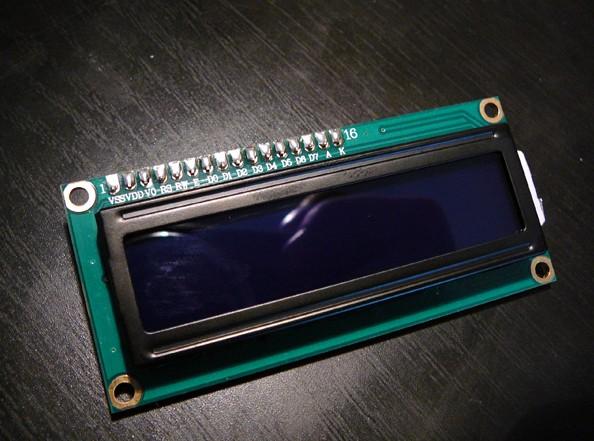


Рис 3.4. Рідкокристалічний дисплей 1602A

Технічні характеристики

* Напруга живлення: 5 В
* Розмір дисплея: 2.6 дюйма
* Тип дисплея: 2 рядки по 16 символів
* Колір підсвічування: синій
* Колір символів: білий
* Габарити: 80мм x 35мм x 11мм

## 3.5. Сервопривід

Сервопривід MG996R 15 кг («Рис. 3.5»)– клон популярного цифрового серводвигуна. Це потужний серводвигун з металевим приводом. Крутний момент до 11 кг / см (при напрузі живлення 6В) [9].



Рис 3.5. Сервопривід MG996R

Характеристики:

* Стабільний та надійний захист від пошкоджень
* Металевий привод
* Підшипник: пластикова втулка
* Довжина проводу 300 мм
* Розміри 40х19х43 мм
* Маса 55 г
* Кут повороту: 120 градусів
* Робоча швидкість: 0.17 сек / 60 градусів (4.8В без навантаження)
* Робоча швидкість: 0.13 сек / 60 градусів (6В без навантаження)
* Пусковий момент: 9.4 кг / см при живленні 4.8В
* Пусковий момент: 11 кг / см при живленні 6В
* Робоча напруга: 4.8 - 7.2В
* Всі деталі приводу виконані з металу

## 3.6. Принципова електрична схема та робочий прототип

Згідно з технічною документацією було скомпоновано робочий прототип розумного акваріуму.

Для отримання розуміння про склад і принцип роботи розумного акваріуму було побудовано принципову електричну схему. Як відомо, цей вид креслення не враховує габаритних розмірів і реального розташування деталей об’єкта, а приділяє основну увагу всім складовим виробу і зв’язкам між ними («Рис.3.6.»).

Рис 3.6. Принципова електрична схема розумного акваріуму

За даною принциповою схемою було поступово складено робочий прототип розумного акваріуму з використанням макетної плати («Рис.3.7»).



Рис 3.7. Прототип розумного акваріуму

Даний прототип було встановлено у справжньому акваріумі і протестовано його роботу. Як показав час – компоненти було підібрано правильно і рішення продемонструвало свою працездатність. Наступним етапом буде створення якісної прошивки та надійного корпусу.

## 3.7 Система відеоспостереження

Система відеоспостереження складається з окремої плати контролера ESP32CAM [18], котра під’єднується до спільної локальної мережі через WiFi. Використання окремого контролера для цієї задачі, обумовлено великим навантаженням через обробку ним відеопотоку. Модуль ESP32CAM був створений саме для трансляцій відеопотоку онлайн, тому існує стандартне програмне забезпечення. Його можливості можна переглянути на «Рис. 3.8».

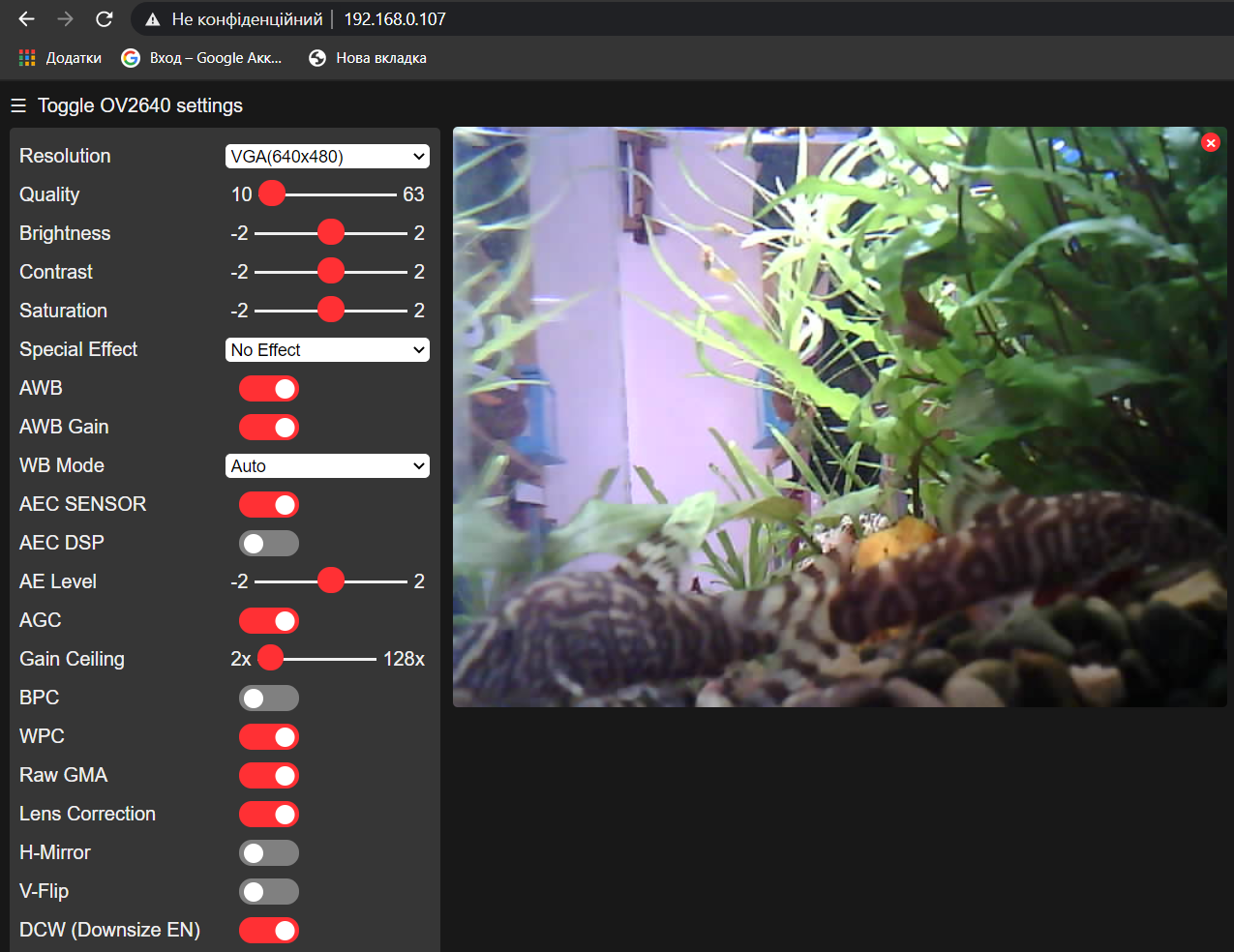


Рис. 3.8 Вигляд роботи веб серверу відеокамери ESP32CAM.

На «Рис. 3.8» у лівій частині видно перелік опцій для зміни параметрів зображення з камери. В залежності від цих налаштувань, можна отримати як плавну картинку в меншому розширені, так і якісне зображення з дуже низьким фреймрейтом. В ході тестування було з’ясовано, що оптимальним розширенням є 640х480 пікселів. А на максимальному розширенні 1600х1200 пікселів модуль контролера дуже сильно нагрівається. Це наштовхує на думку, що для такої камери потрібно застосовувати додатковий радіатор охолодження.

# РОЗДІЛ 4

# Програмний код

## 4.1. Веб-сайт для керування

Однією з найважливіших цілей у проєкті було створення вебсайту для дистанційного керування розумним акваріумом [10]. На сьогодні сайт виглядає так як зображено на «Рис. 4.1».

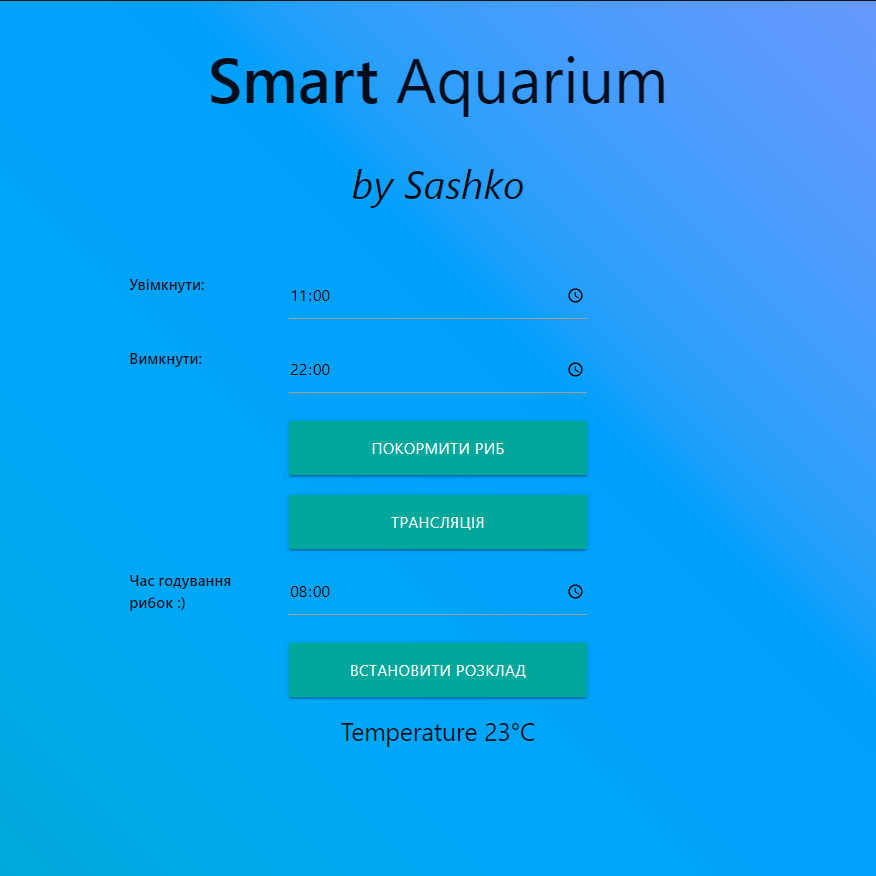


Рис 4.1. Сторінка керування розумним акваріумом

Було використано HTML та форматування за допомогою звичайних стилів [11].

На сайті користувач можете вказати такі параметри: час сонячного дня та час запуску корму риб. Також користувач можете дізнатися про температуру у акваріумі та етап сонячного дня (ранок\день\вечір\ніч).

У майбутньому планується також підключення трансляції з вбудованої камери ESP32, щоб можна було спостерігати за улюбленцями, незважаючи на відстані.

## 4.2. Прошивка мікроконтролера

У ході виконання роботи було написано програмний код для мікроконтролера для опрацьовування команд з веб-сайту та керуванням всіма периферійними пристроями [5, 12–14]. Для створення веб серверу було обрано бібліотеку ESPAsyncWebServer, котра забезпечує асинхронну обробку запитів від користувачів, і таким чином дозволяє використання веб інтерфейсу багатьма користувачами одночасно. Також, програма використовує принцип таймерів для виконання усіх функцій пристрою, завдяки чому мікроконтролер не очікує повного виконання одного завдання, і завжди готовий виконувати наступне.

Пройдемося по основних моментах коду та надамо пояснення.

|  |
| --- |
| #include <**WiFi**.h>  #include "FS.h"  #include "SPIFFS.h"  #include <OneWire.h>  #include <DallasTemperature.h>  #include <ESP32Servo.h>  #include <ESPAsyncWebServer.h>  #include <AsyncTCP.h>  #include <Arduino.h>  #include "MillisTimer.h"  #include "RTClib.h"  #include <Wire.h>  #include "timeParser.h"  #include "js.h" |

Підключили потрібні бібліотеки для роботи з файловою системою, сервоприводом [15], датчиком температури, модулем WiFi, реле [16], таймерів ,тощо.

|  |
| --- |
| RTC\_DS1307 rtc;  char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};  **Servo** servo;  String web,header;  // Config WiFi parameters  const char\* ssid = "1dlink";  const char\* password = "1111111111";  **IPAddress** staticIP(192,168,43,122);  **IPAddress** gateway(192,168,43,1);  **IPAddress** subnet(255,255,255,0); |

Створили об’єкти для давача реального часу та сервоприводу. Вказали параметри підключення до мережі WiFi - задали ІР адреси та маску підмережі.

|  |
| --- |
| // Init FORM DATA parameters  const char\* PARAM\_INPUT\_1 = "vkl";  const char\* PARAM\_INPUT\_2 = "vukl";  const char\* PARAM\_INPUT\_3 = "zzz";  String time\_vkl; // "08:00"  String time\_vukl;  String time\_zzz;  MillisTimer time\_timer = MillisTimer(59999);  bool motor\_run = false;  bool anti\_double\_feed  = false;  MillisTimer timer\_anti\_double\_feed = MillisTimer(43200000);  bool light\_status = false;  int light\_0\_256 = 0;  int temp = 0;  MillisTimer light\_timer = MillisTimer(1000); |

Оголосили змінні, що будуть використовувати при обчисленні часових інтервалів для годування рибок і підсвітки акваріуму, а також створили необхідні таймери. Вказали значення за замовчуванням.

|  |
| --- |
| //D18B20 temperature  OneWire oneWire(ONE\_WIRE\_BUS);  DallasTemperature sensors(&oneWire);  MillisTimer temperature\_timer = MillisTimer(feed\_time); |

Оголосили змінні, що будуть використовувати при роботі з давачем температури і таймер для виміру температури.

|  |
| --- |
| void setup() {   //d18b20   sensors.begin();    **Serial**.begin(115200);      servo.attach(15);                   //servo      pinMode(2, OUTPUT);                 //mosfet    digitalWrite(2, LOW);  **Serial**.println("Prepare pins");   pinMode(I2C\_SCL,INPUT\_PULLUP);//SDA // ds1307rtc   pinMode(I2C\_SDA,INPUT\_PULLUP);//SCL  **Serial**.println("Prepare pins Done");   Wire.begin(I2C\_SCL,I2C\_SDA);  **Serial**.println("I2C up and running");     if (!rtc.begin()) {  **Serial**.println("Couldn't find RTC");     abort(); }   if (!rtc.isrunning()) {  **Serial**.println("RTC is NOT running, let's set the time!");     rtc.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));}      if(!**WiFi**.config(staticIP, gateway, subnet)){  **Serial**.println("Cant configure wifi!");}  **Serial**.println("Prepare WiFi");  **WiFi**.begin(ssid, password);          //WiFi    while (**WiFi**.status() != WL\_CONNECTED) {     delay(500);  **Serial**.print(".");    }  **Serial**.println("Connected to Wifi"); |

Функція “setup”, використовується для ініціалізації периферійного обладнання. В ній відбувається підключення до мережі WiFi.

|  |
| --- |
| //file system read   if(!**SPIFFS**.begin()){  **Serial**.println("An Error has occurred while mounting SPIFFS");     return;   }   File file = **SPIFFS**.open("/index.html", "r");  **Serial**.println();  **Serial**.println("File Content:");   delay(2000);   while (file.available())   {   web+=String((char)file.read());   }   file.close();       temperature\_timer.expiredHandler(measure\_temp);   temperature\_timer.setInterval(feed\_time);   temperature\_timer.setRepeats(1);   temperature\_timer.start();   time\_timer.expiredHandler(timer\_ds);   time\_timer.setRepeats(1);   time\_timer.start();   light\_timer.expiredHandler(lightUpdate);   light\_timer.setRepeats(1);   light\_timer.start(); |

В цій самій функції відбувається налаштування файлової системи, та початковий запуск таймерів виміру температури, звірки з розкладом годування та підсвічування, таймер підсвітки.

Далі необхідно було зареєструвати події, на які буде реагувати веб сервер. Передбачено обробку GET та POST запитів.

|  |
| --- |
| server.on("/", HTTP\_GET, [] (AsyncWebServerRequest \*request){       String html\_send = web +  showTemp(temp)+ feedButton(anti\_double\_feed);       request->send(200, "text/html", html\_send + setRozklad(time\_vkl, time\_vukl, time\_zzz));    }); |

Реєструється подія звертання до головної сторінки. Формується HTML сторінка з JavaScript скриптами що повертаються функціями showTemp(), feedButton() та setRozklad(). Перший скрипт підставляє на сторінку оновлене значення температури, а другий визначає чи буде активною кнопка “погодувати рибок”, адже програма має захист від їх перегодовування.

|  |
| --- |
| server.on("/feed\_fish", HTTP\_GET, [] (AsyncWebServerRequest \*request){       if (!anti\_double\_feed) {         motor\_run = true;         anti\_double\_feed = true;         feed\_timer.expiredHandler(turn\_off\_motor);         feed\_timer.setInterval(feed\_time);         feed\_timer.setRepeats(1);         feed\_timer.start();         timer\_anti\_double\_feed.expiredHandler(security\_watchdog);         timer\_anti\_double\_feed.setInterval(43200000);         timer\_anti\_double\_feed.setRepeats(1);         timer\_anti\_double\_feed.start();}       String html\_send = web +  showTemp (temp) + setRozklad(time\_vkl, time\_vukl, time\_zzz) + feedButton(anti\_double\_feed);       request->send(200, "text/html", html\_send);    }); |

За адресою “/feed\_fish” реєструється логіка годування в акваріумі. Програма не дозволить годувати кілька разів підряд, як це до прикладу, можуть зробити маленькі діти, проте, якщо перевірку буде пройдено –запускається двигун що обертає шнек [15, 16], і корм дозовано потрапляє до акваріуму.

|  |
| --- |
| server.on("/get", HTTP\_POST, [] (AsyncWebServerRequest \*request) {         String inputMessage;         String inputParam;         if (request->hasParam(PARAM\_INPUT\_1, true)) {           inputMessage = request->getParam(PARAM\_INPUT\_1, true)->value();           time\_vkl=inputMessage;  **Serial**.println("Morning: "+time\_vkl);           }         if (request->hasParam(PARAM\_INPUT\_2, true)) {           inputMessage = request->getParam(PARAM\_INPUT\_2, true)->value();           time\_vukl=inputMessage;  **Serial**.println("Evening: "+time\_vukl);         }         if (request->hasParam(PARAM\_INPUT\_3, true)) {           inputMessage = request->getParam(PARAM\_INPUT\_3, true)->value();           time\_zzz=inputMessage;  **Serial**.println("Nyam nyam)"+time\_zzz);         }         String html\_send = web + showTemp(temp) + feedButton(anti\_double\_feed);         request->send(200, "text/html", html\_send+setRozklad(time\_vkl, time\_vukl, time\_zzz));    });  server.begin();    } |

Реєструється обробник події отримання даних нового розкладу з форми, яку заповнив користувач. Якщо такі дані були прийняті, то відповідні глобальні змінні для часу оновлюються. Вкінці запускаємо веб сервер, на чому його налаштування і функція “setup” завершуються.

|  |
| --- |
| void loop() {   motor\_run = false;   temperature\_timer.run();   feed\_timer.run();   time\_timer.run();   light\_timer.run();   timer\_anti\_double\_feed.run();   if (motor\_run ==true){       servo.write(0);  **Serial**.println("Motor ON");     }  } |

Це основний цикл програми розумного акваріума. Тут здійснюється оновлення всіх таймерів програми.

Опишемо функції котрі викликають ці таймери після закінчення часу їх роботи:

|  |
| --- |
| void measure\_temp(MillisTimer &mt){   sensors.requestTemperatures();   temp = sensors.getTempCByIndex(0);  **Serial**.println("Tempereture measurment:" + String(temp));   mt.reset();   mt.setInterval(feed\_time);   mt.expiredHandler(measure\_temp);   mt.setRepeats(1);   mt.start();  } |

Функція вимірює температуру води акваріуму та перезапускає таймер.

|  |
| --- |
| void lightUpdate(MillisTimer &mt){     if (light\_status) {         if (light\_0\_256 <255){              light\_0\_256+=8;}     } else {         if (light\_0\_256 > 0){              light\_0\_256-=8;}     }  **Serial**.println("Analog points: " + String(light\_0\_256));     analogWrite(2, light\_0\_256);     mt.setRepeats(1);     mt.start();  } |

Функція плавно регулює освітлення в акваріумі, в залежності від розкладу інтенсивність освітлення збільшується чи зменшується.

|  |
| --- |
| void security\_watchdog(MillisTimer &mt){     anti\_double\_feed  = false;} |

Оновлює прапорець повторного годування кожні 12 годин.

|  |
| --- |
| void turn\_off\_motor(MillisTimer &mt){   servo.write(90);   motor\_run = false;   feed\_timer.reset();  **Serial**.println("Motor OFF");   servo.write(90);   } |

Дана функція вимикає двигун, після закінчення процесу годування.

|  |
| --- |
| void timer\_ds(MillisTimer &mt){     DateTime now = rtc.now();       //Time variables     String vkl\_hour, vkl\_minute = "";   // "08", "00"     String vukl\_hour, vukl\_minute = "";     String zzz\_hour, zzz\_minute = "";     zzz\_hour = getValue(time\_zzz, ':', 0);     zzz\_minute = getValue(time\_zzz, ':', 1);     vkl\_hour = getValue(time\_vkl, ':', 0);     vkl\_minute = getValue(time\_vkl, ':', 1);     vukl\_hour = getValue(time\_vukl, ':', 0);     vukl\_minute = getValue(time\_vukl, ':', 1);     if (zzz\_hour.charAt(0)=='0'){zzz\_hour = zzz\_hour.charAt(1);}     if (zzz\_minute.charAt(0)=='0'){zzz\_minute = zzz\_minute.charAt(1);}     if (vkl\_hour.charAt(0)=='0'){vkl\_hour = vkl\_hour.charAt(1);}     if (vkl\_minute.charAt(0)=='0'){vkl\_minute = vkl\_minute.charAt(1);}     if (vukl\_hour.charAt(0)=='0'){vukl\_hour = vukl\_hour.charAt(1);}     if (vukl\_minute.charAt(0)=='0'){vukl\_minute = vukl\_minute.charAt(1);}       if ((String)now.hour()==zzz\_hour && (String)now.minute()==zzz\_minute){  **Serial**.println(zzz\_hour+" "+zzz\_minute);         motor\_run =true;         feed\_timer.expiredHandler(turn\_off\_motor);         feed\_timer.setInterval(feed\_time);         feed\_timer.setRepeats(1);         feed\_timer.start();}  **Serial**.print(now.hour() + vkl\_hour);     if ((String)now.hour()==vkl\_hour && (String)now.minute()==vkl\_minute){       light\_status = true;       }     if ((String)now.hour()==vukl\_hour && (String)now.minute()==vukl\_minute){       light\_status = false;       }   mt.setRepeats(1);  mt.start();  } |

Функція перевірки розкладу. Тут міститься основна логіка, що запускає процеси у зазначений в розкладі час.

Також до основного коду програми прикріплюються 2 власні бібліотекі, які містять функції для формування JavaScript скриптів та парсингу часу зі стрічки. (**js.h** та **timeParser.h**)

Бібліотека **js.h**

|  |
| --- |
| String setRozklad(String time\_vkl, String time\_vukl, String time\_zzz ) {     String s = String("<script> let input\_vkl = document.getElementById(\"vkl\");")  +              "let input\_vukl  = document.getElementById(\"vukl\");" +              "let input\_zzz  =document.getElementById(\"zzz\");" +              " if (\"" + time\_vkl + "\"!= \"\") {input\_vkl.value =  \"" + time\_vkl + "\";}" +              " if (\"" + time\_vukl + "\" != \"\") {input\_vukl.value  = \"" + time\_vukl + "\"}" +              " if (\"" + time\_zzz + "\" != \"\") {input\_zzz.value  =\"" + time\_zzz + "\"}" +              "</script>";   return s;  }  String showTemp (byte temp) {   return "<script>let elem = document.querySelector(\".temp\");elem.innerHTML = \"Temperature " + String((int)temp) + "°С\";</script>";  }  String feedButton (bool anti\_double\_feed){   String buttonClass ="";   if (anti\_double\_feed){     buttonClass = "btn-large disabled waves-effect waves-light";     }   else{     buttonClass = "btn-large waves-effect waves-light";   }   return "<script>let elem = document.querySelector(\"#feedButton\"); elem.className=\""+ buttonClass +"\"; </script>";    } |

У цій бібліотеці ми створюємо 3 окремих функції Java Script (setRozklad, showTemp, feedButton). Функція setRozklad() встановлює значення розкладу(вмикання/вимикання світла в акваріумі та час годування рибок) в поля вводу форми HTML. showTemp() оновлює значення температури на веб-сторінці. А функція feedButton() перефарбовує кнопку на веб-сторінці у сірий колір, та дизактивує її.

Бібліотека **timeParser.h**

|  |
| --- |
| String getValue(String data, char separator, int index)  {     int found = 0;     int strIndex[] = { 0, -1 };     int maxIndex = data.length() - 1;     for (int i = 0; i <= maxIndex && found <= index; i++) {         if (data.charAt(i) == separator || i == maxIndex) {             found++;             strIndex[0] = strIndex[1] + 1;             strIndex[1] = (i == maxIndex) ? i+1 : i;         }     }     return found > index ? data.substring(strIndex[0], strIndex[1]) : "";  } |

У цій бібліотеці ми створюємо парсер-функцію, яка розбиває стрічку з часом по символу двокрапки (“:”), та використовуємо її у основному коді програми.

# РОЗДІЛ 5

# Проектування 3D моделей елементів корпусу

Всі деталі корпусу та кріплення, які використовувалися у проєкті, були створенні у програмі **SolidWorks 2019** та надруковані на 3D-принтері [17]. Відповідні ескізні рисунки деталей наведені нижче:

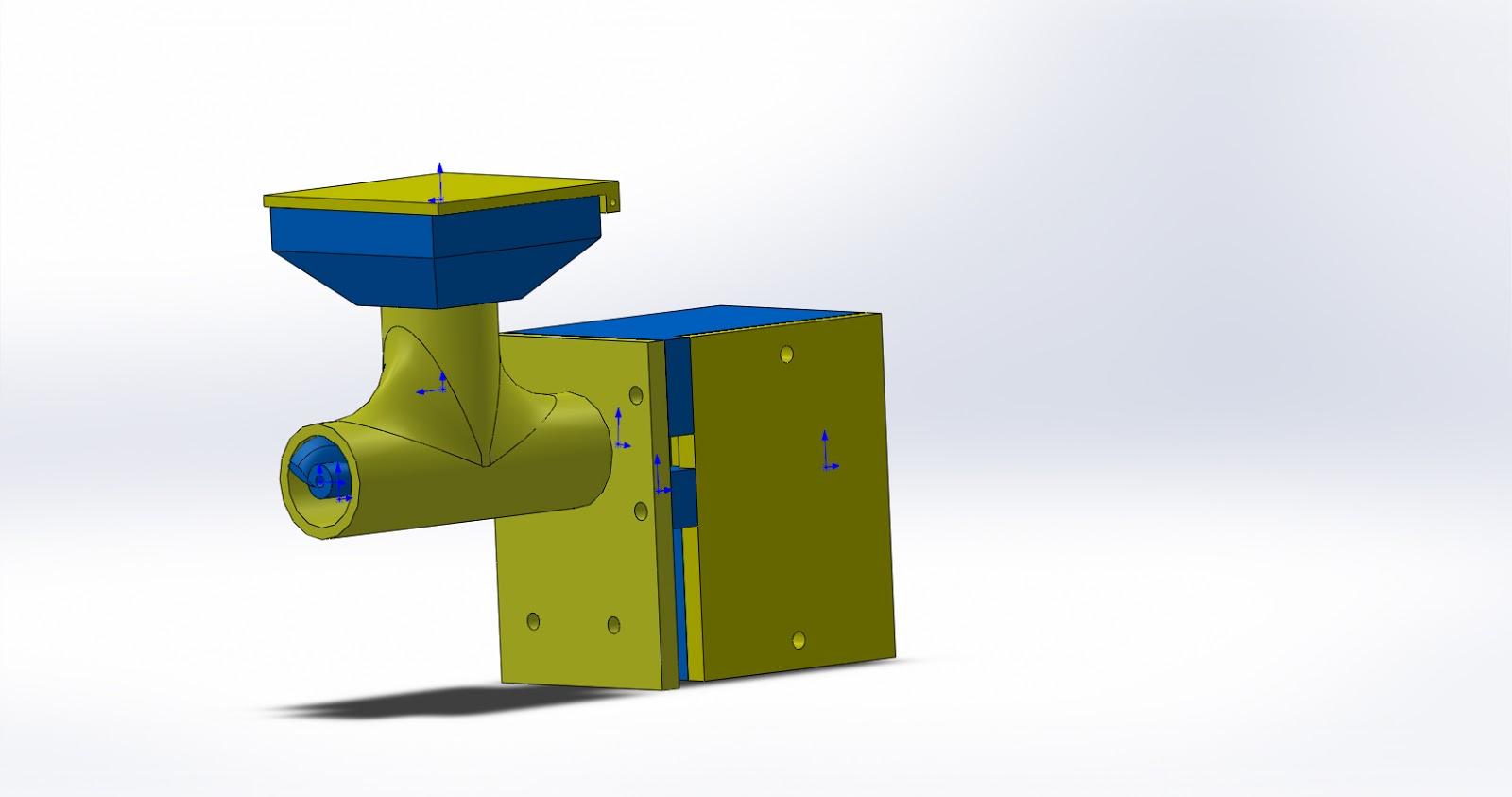
Електрогодівничка «Рис.5.1»****

Рис 5.1. Електрогодівничка

Електрогодівника складається з 6 деталей:

1. Шнек
2. Трубка-з’єднання
3. Коробка для двигуна
4. Кришка для двигуна
5. Кормовий танк
6. Кришка для танка

Шнек «Рис.5.2»

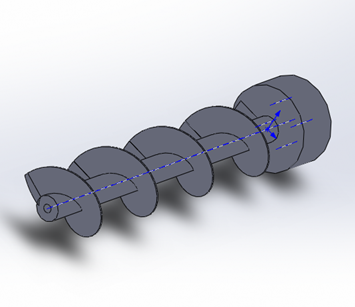


Рис 5.2. Шнек

Трубка-з’єднання «Рис.5.3»

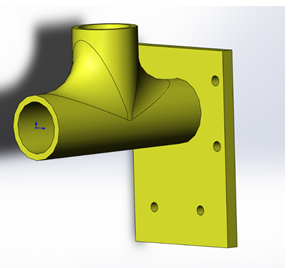


Рис 5.3. Трубка-з’єднання

Коробка для двигуна «Рис.5.4»

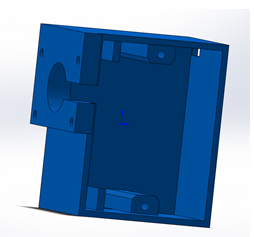


Рис 5.4. Коробка для двигуна

Кришка для двигуна «Рис.5.5»

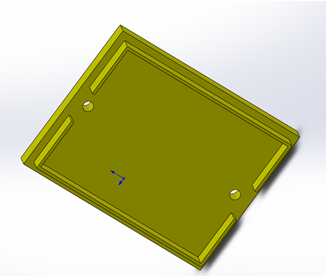


Рис 5.5. Кришка для двигуна

Кормовий танк «Рис. 5.6»

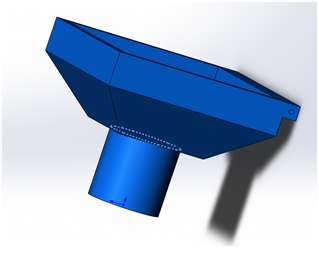


Рис 5.6. Кормовий танк

Кришка для танку «Рис.5.7»

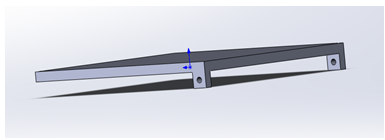


Рис 5.7 . Кришка для танка

# ВИСНОВОК

Розглянувши ринок подібних розумних систем, було з’ясовано, що середня ціна таких пристроїв є високою. Але на сьогодні доступні дешеві контролери на базі яких можна розробити розумний акваріум власноруч.

У ході виконання даної наукової роботи був розроблений власний програмно-апаратний комплекс, що надає користувачу інформацію про стан акваріумного середовища на смартфон у режимі реального часу та має зрозумілий інтерфейс. За допомогою веб-інтерфейсу можна запланувати сонячний день в акваріумі, а також розклад харчування рибок. Також є можливість переглянути камеру відеоспостереження і годувати рибок вручну, контролюючи процес через Інтернет. У програмному забезпечені було використано асинхронний веб-сервер, який дозволяє користуватися пристроєм багатьом клієнтам одночасно. Був продуманий і реалізований захист від багаторазового годування риб.

У роботі над апаратною частиною було реалізовано не тільки поєднання, налаштування та тестування роботи різних електронних компонентів, а й здійснено проектування та 3D друк корпусу. З метою зменшення кількості рухомих частин було обрано схему подачі корму шнеком. Це позитивно вплинуло на надійність та дешевизну конструкції. Вибрана апаратна частина надає можливість підключення різноманітних датчиків, а це значно розширює межі використання створеного продукту.

Розроблена комп’ютерна система має перспективи для використання у навчальних цілях, а також для власних досліджень. У майбутньому планується інтеграція усіх компонентів системи в єдиний компактний корпус з використанням друкованих плат. Такі удосконалення дозволять перевести проєкт у комерційну площину.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jin: Xiaomi представила умный аквариум с подсветкой и Wi-Fi. iXBT.com, 7 August 2020. https://www.ixbt.com/news/2020/08/07/xiaomi-wi-fi.html. Accessed 14 February 2021.854Z

2. GizChina.it: Аквариум Xiaomi YouPin занимает poco место и стоит 35 евро - GizChina.it. https://ru.gizchina.it/2020/12/xiaomi-youpin-sobo-%D0%B0%D0%BA%D0%B2%D0%B0%D1%80%D0%B8%D1%83%D0%BC/ (2020). Accessed 14 February 2021.273Z

3. Aquael Shrimp Set Smart. https://zoo-baza.com.ua/shrimp-set-smart.html (2021.000Z). Accessed 14 February 2021.610Z

4. 1wire.com.ua: Модуль ESP32-CAM с камерой OV2640 2MP - Купить в Украине, Харьков. https://1wire.com.ua/esp32-cam.html (2021.000Z). Accessed 14 February 2021.041Z

5. Hoddie, L.P.P., Prader, L.: IoT Development for ESP32 and ESP8266 with JavaScript. Apress (2020)

6. Датчик влажности и температуры DS18B20: описание, подключение, схема, характеристики | ВИКИ. https://3d-diy.ru/product/germetichnyy-datchik-temperatury-ds18b20. Accessed 14 February 2021.436Z

7. RobotChip: Обзор часов реального времени DS1307 (RTC) – RobotChip. https://robotchip.ru/obzor-chasov-realnogo-vremeni-ds1307/ (2016). Accessed 14 February 2021.196Z

8. RobotChip: Обзор LCD-дисплея 1602A (LCD1602A) – RobotChip. https://robotchip.ru/obzor-lcd-displeya-1602a/ (2016). Accessed 14 February 2021.179Z

9. Arduino в Украине: Сервопривід MG996R 15 кг в Києві та Україні. https://arduino.ua/prod272-servoprivod-mg996r-15-kg (2021.000Z). Accessed 14 February 2021.758Z

10. Build an ESP8266 Web Server with Arduino IDE - Code and Schematics - YouTube. https://www.youtube.com/watch?v=dWM4p\_KaTHY&ab\_channel=RuiSantos (2021.000Z). Accessed 14 February 2021.530Z

11. Color - Materialize. materializecss.com (2021.000Z). Accessed 14 February 2021.610Z

12. Блум, Д.: Изучаем Arduino: инструменты и методы технического волшебства. БХВ-Петербург (2015)

13. Воронин, И., Воронина, В.В.: Программирование для детей. От основ к созданию роботов. СПб.: Питер (2018)

14. Diy Fish Feeder | Arduino project # 7 | MakerMan - YouTube. https://www.youtube.com/watch?v=Lw-AT77I3GU (2021.000Z). Accessed 14 February 2021.366Z

15. Уроки Arduino - управление сервомашинкой - YouTube. https://www.youtube.com/watch?v=S0tb3jnfdw4&t=212s&ab\_channel=%D0%97%D0%B0%D0%BC%D0%B5%D1%82%D0%BA%D0%B8%D0%90%D1%80%D0%B4%D1%83%D0%B8%D0%BD%D1%89%D0%B8%D0%BA%D0%B0 (2021.000Z). Accessed 14 February 2021.865Z

16. IOT - Building Reliability: How the 5v relay works ?? https://mounishkokkula.wordpress.com/how-the-5v-relay-works/ (2017). Accessed 14 February 2021.185Z

17. SolidFactory. Видеоуроки SolidWorks - YouTube. https://www.youtube.com/channel/UC4fc5wHqEoY3Ro3mu2IUOew (2021.000Z). Accessed 14 February 2021.541Z

18 ESP32-CAM ov2640, потоковое видео в среде Arduino IDE. https://portal-pk.ru/news/228-esp32-cam-ov2640-potokovoe-video-v-srede-arduino-ide.html. Accessed 21 February 2021.