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 ІТ - школи

Рівненського міського

Палацу дітей та молоді

учень ЗЗСО №1

Чернюк Олександр Олександрович

Науковий керівник :

Ольховик Вадим Леонідович

керівник гуртка

профільної ІТ школи

Рівненського міського

Палацу дітей та молоді

Рівне 2021

**Назва роботи:** Розумний акваріум

**Виконав:** Чернюк Олександр Олександрович.

**Територіальне відділення Малої академії наук України:** Рівненська Мала

академія наук учнівської молоді.

**Базовий позашкільний навчальний заклад:** Рівненський міський Палац дітей та молоді.

**Науковий керівник:** Ольховик Вадим Леонідович, керівник гуртків ПДМ.

**Основна мета роботи:** Розробити пристрій, який полегшить користувачам догляд за акваріумних середовищем.

**Актуальність:** Розвиток Інтернету речей в побутовій сфері.

**Завдання наукового дослідження:** Розглянути існуючі на ринку пристрої для автоматичного годування акваріумних рибок. Спроектувати оптимальну конструкцію подібного пристрою, та розробити зручний інтерфейс.

**Мова програмування:** Arduino, HTML5\CSS.

**Використані сторонні бібліотеки:** WiFi.h, SimpleDHT, LiquidCrystal\_I2C, FS.h, SPIFFS.h, Servo.h.

**Висновок:** проведено дослідження існуючих методів  дистанційного управління та їх застосувань. Створено програму, яка може використовуватися у власних досліджень акваріумного середовища і  спростити догляд за рибами в акваріумі.

**ЗМІСТ**

[Вступ 4](#_Toc61209297)

[РОЗДІЛ 1 Що таке Смарт-акваріум і як він працює? 5](#_Toc61209298)

[РОЗДІЛ 2 Подібні проекти 7](#_Toc61209299)

[РОЗДІЛ 3 Використанні модулі, та їх роль у проекті 9](#_Toc61209300)

[РОЗДІЛ 4 Програмний код 10](#_Toc61209301)

[РОЗДІЛ 5 3D моделі 16](#_Toc61209302)

[Висновок 19](#_Toc61209303)

[Список використаних джерел 20](#_Toc61209304)

# ВСТУП

У сучасному світі програмування застосовується для вирішення завдань широкого спектру. Нині багато передових компаній створюють власні додатки, які можуть контролювати предметами, які знаходяться у будинку, дистанційно, а іноді взагалі й без вашої потреби. Можемо взяти до прикладу smart-house, який включає у себе різний асортимент електричних приладів у вашому будинку. Поки повертаєтесь з роботи, Ви можете за допомогою смартфону включити *електрочайник* та *робот-порохотях,* та до приїзду на Вас вже чекатиме майже готовий теплий чай та чиста й прибрана кімната. Таких приладів може бути безліч, починаючи від холодильників та пральних машинок й закінчуючи звичайно. електро годівничкою для тварин.

Дивлячись на це все, з’являться запитання:  “Чи є взагалі такі проекти як smart-house, тільки для акваріуму?” Над цим проєктом йде робота близько двох років. Минулого року була ціль створити штучну імітацію сонячного світла за допомогою LED-стрічки. А ціль, яка стоїть нині – це зробити електрогодівничку для риб, та створити сайт, на якому можна було б дистанційно спостерігати за водним середовищем. Розробити 3D-макет станції та надрукувати її на 3D-принтері.

Перед тим як приступити до проєктування, було розглянуто ринок подібних пристроїв, ціна яких нам була недовподоби, 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,який приваблює нас своєю флорой і фауной. Цікаво спостерігати за плаванням риб, ростом рослин, пристосування організмів до нового середовища, і не тільки. Також, акваріум є непоганим елементом декору, яким можна прикрасити свою кімнату. Але, як і для кожної домашньої тварини, за акваріумним середовищем потрібен ретельний догляд, само собою догляд за акваріумом є досить-таки складною процедурою. Тому спрощення догляду за акваріумом і стало метою наукової роботи.

Проєкт Смарт-акваріум налічує у собі відразу декілька корисних фунцій: LED-підсвітка з імітацією сонячного світла, електрогодівничка, камера відеоспостереження, Wi-Fi керування, датчик температури і не тільки.

Перевагою цього проєкту є у тому, що головна конструкція живиться від звичайної 5-вольтової зарядки телефону, яка є майже у кожному будинку, але є одне застереження. Сила струму в такій зарядці повинна бути не менше 2 Ампер. Для прикладу, можна взяти звичайну зарядку від компанії Xiaomi, яка і стала в нагоді під час розробки даного проєкту.

Імітація сонячного світла відбувається завдяки LED-стрічки, яка живиться від мережі 5 вольт і керується ШІМ-сигналом. Завдяки платі ESP32, яка 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 простоти підрахунків, я обмежив бітність АЦП до 10-ти біт. Тобто, зараз є 1024 рівнів яскравості, де 0 - це виключення стрічки, а 1023 - це її повна потужністю.

Електрогодівничка стала однією з найважчих цілей в даному проєкті. Повне моделювання та друк всіх елементів зайняло близько декількох тижнів. Зверху цієї конструкції є мала спеціальна ємність для корму, який потрапляє в акваріум завдяки руху шнека. Ця деталь, в свою чергу, обертається завдяки встановленому серво-приводу.  Перевагою цієї годівнички є в тому, що корм подається кожного дня в певну задану годину, незалежно від нашої присутності. Також, якщо нам хочеться знову погодувати риб, ми це можемо зробити завдяки кнопці, яка розміщена на веб інтерфейсі контролера.

Всі ці дії контролює одна плата ESP32,  в якій присутній Wi-Fi модуль, за допомогою якого ми і хостимо сайт. В 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 проекти

Насправді, подібних проєктів на ринку не дуже багато. Невеликий попит на такі середовища, адже інколи їхня ціна сягає від 75-600 доларів США. Ціна залежить від об’єм самого акваріума. Лідерами серед продаж, як не дивно,  є компанія Xiaomі з своїми **Fish Tank Pro**«**рис 2.1**» **і YouPin** «**рис 2.2** »**,** та компанія AqueEl з **SHRIMP SET SMART**«**рис 2.3**»**.** Хоча, на ринку існують менш відомі виробники, які також не відступають від лідерів розумних акваріумів, створюючи нові функції та можливості з хорошою якістю, та гарним дизайном за приємну ціну, такі, як: **Hoison**«**рис 2.4**», **Silhouette Tetra** «**рис 2.5**»**,** та багато інших...

Ці проєкти Ви можете переглянути за допомогою QR-коду

**Рис 2.1 Fish Tank Pro**

**Рис 2.2 YouPin**



**Рис 2.3 SHRIMP SET SMART**

**Рис 2.4 Hoison**



**Рис 2.5 Silhouette Tetra**

****

# РОЗДІЛ 3 Використанні модулі, та їх роль у проекті

Під час розробки цього проєкту використовувалися такі модулі як:

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

Бібліотеки, які були використані під час розробки проєкту:

1. Servo.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)

Також хотілося б виділити 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, та має непогану потужність. Якщо її порівнювати з платами Arduino, такими як: Uno, Nano, Leo, Wemos та багато інших, то ESP32 буде лідером за тактовою частотою, та об’ємом пам’яті. Якщо вже говорити про потужність, то для даного проєкту непоганим варіантом була б плата Raspberry Pi, але її ціна кілька разів перевищує ціну обраного контролера. Тому ESP32, маючи невеликі розміри та оптимальну ціну, стає прекрасним вибором для даного проєкту.

# РОЗДІЛ 4 Програмний код

Однією з найважливіших цілей у проєкті є створення веб-сайту для дистанційного керування. На даний момент сайт виглядає наступним чином.



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

У майбутньому планується також підключення транслації з вбудованої камери ESP32, щоб ви могли спостерігати за вашими улюбленцями незважаючи на відстані.

# #include <WiFi.h>

# #include "FS.h"

# #include "SPIFFS.h"

# #include <SimpleDHT.h>

# #include <ESP32Servo.h>

# Servo servo;

# const char\* ssid = "dlink";

# const char\* password = "1111111111";

# String web,header;

# WiFiServer server(80);

# IPAddress staticIP(192,168,0,22);

# IPAddress gateway(192,168,0,1);

# IPAddress subnet(255,255,255,0);

# bool client\_flag = true;

# unsigned long int feed\_time = 3000;

# unsigned long int current = millis();

# unsigned long currentTime = millis();

# // Previous time

# unsigned long previousTime = 0;

# // Define timeout time in milliseconds (example: 2000ms = 2s)

# const long timeoutTime = 2000;

# int pinDHT11 = 14;

# SimpleDHT11 dht11(pinDHT11);

# byte temp = 0;

# byte hum = 0;

# 

# void setup() {

# servo.attach(15);

# pinMode(2, OUTPUT);

# digitalWrite(2, LOW);

# WiFi.config(staticIP, gateway, subnet);

# Serial.begin(115200);

# WiFi.begin(ssid, password);

# while (WiFi.status() != WL\_CONNECTED) {

# delay(500);

# 

# Serial.print(".");

# Serial.println("");

# Serial.println("WiFi connected.");

# Serial.println("IP address: ");

# Serial.println(WiFi.localIP());

# server.begin();

# 

# 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();

# }

# Serial.println(web);

# }

# 

# void loop() {

# servo.write(0);

# Serial.println("woto");

# WiFiClient client = server.available(); // прослушка входящих клиентов

# if (client&&client\_flag) {

# client\_flag = false;

# currentTime = millis();

# previousTime = currentTime;

# Serial.println("New Client."); // выводим сообщение

# String currentLine = "";

# while (client.connected() && currentTime - previousTime <= timeoutTime ) {

# //currentTime = millis();

# if (client.available()) { // если от клиента поступают данные,

# char c = client.read(); // читаем байт, затем

# Serial.write(c); // выводим на экран

# header += c;

# if (c == '\n') { // если байт является переводом строки

# // если пустая строка, мы получили два символа перевода строки

# // значит это конец HTTP-запроса, формируем ответ сервера:

# if (currentLine.length() == 0) {

# Serial.println(header);

# if(header.indexOf("GET /feed\_fish")>=0){

# servo.write(180);

# digitalWrite(2, HIGH);

# current = millis();

# Serial.println("Включення двигуна");

# }

# // HTTP заголовки начинаются с кода ответа (напр., HTTP / 1.1 200 OK)

# // и content-type, затем пустая строка:

# client.println("HTTP/1.1 200 OK");

# client.println("Content-type:text/html");

# client.println("Connection: close");

# client.println();

# 

# client.println(web+" <div class=\"row\"><h5>Temperature "+temp+"°С</h5> </div></div></body></html>");

# client.println("\n\r");

# delay(5);

# break;

# }

# else { // если получили новую строку, очищаем currentLine

# currentLine = "";

# }

# }

# else if (c != '\r') { // Если получили что-то ещё кроме возврата строки,

# currentLine += c; // добавляем в конец currentLine

# }

# }

# }

# header = "";

# 

# client.stop();

# delay(1000);

# Serial.println("Client disconnected.");

# Serial.println("");

# }

# if (millis()-current>=feed\_time){

# servo.write(90);

# Serial.println("Виключення двигуна");

# digitalWrite(2, LOW);

# client\_flag=true;

# dht11.read(&temp, &hum, NULL);

# 

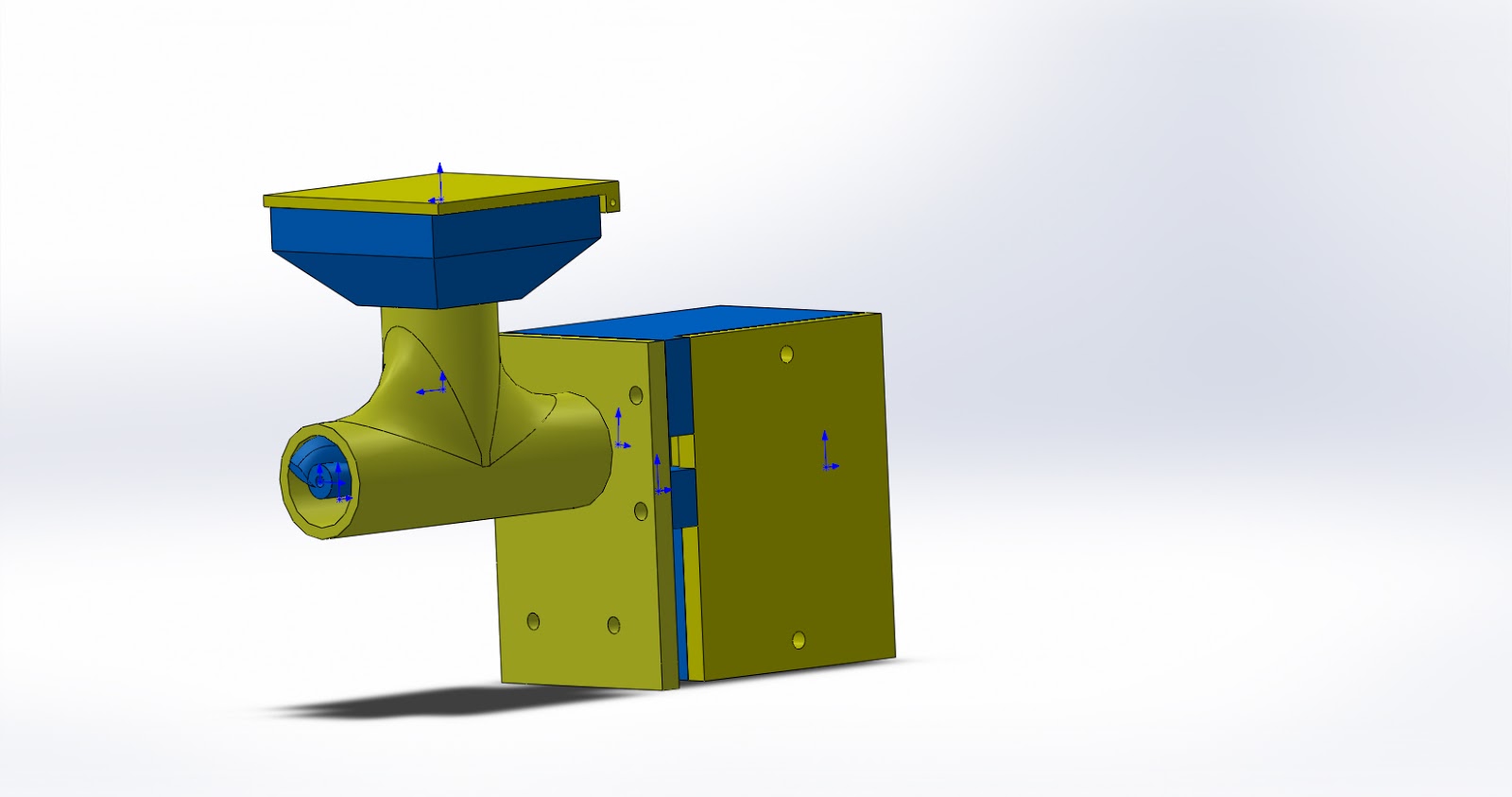
# }

# } РОЗДІЛ 5 3D моделі

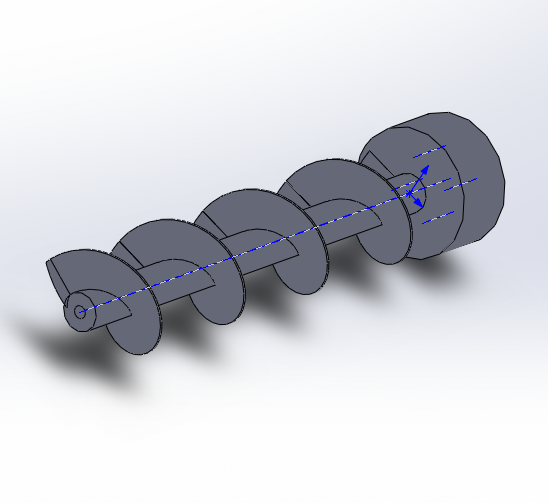
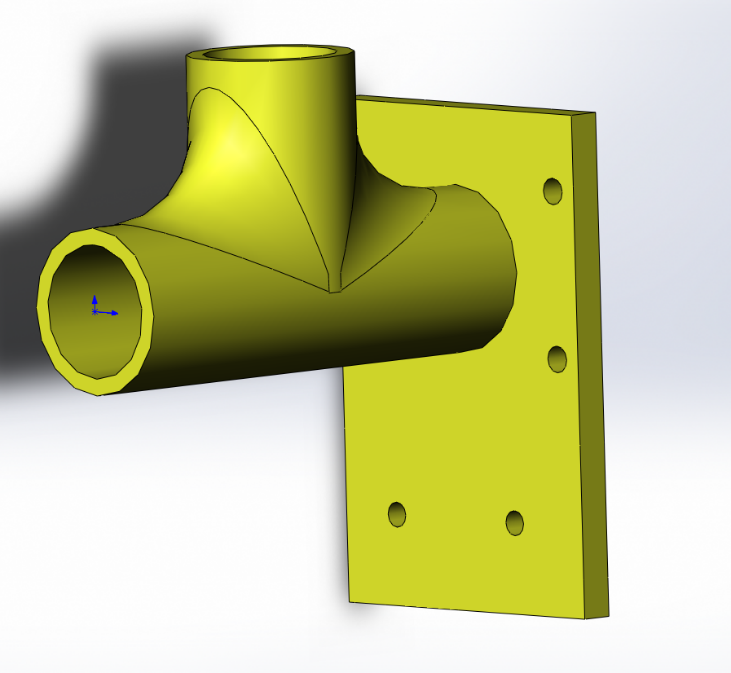
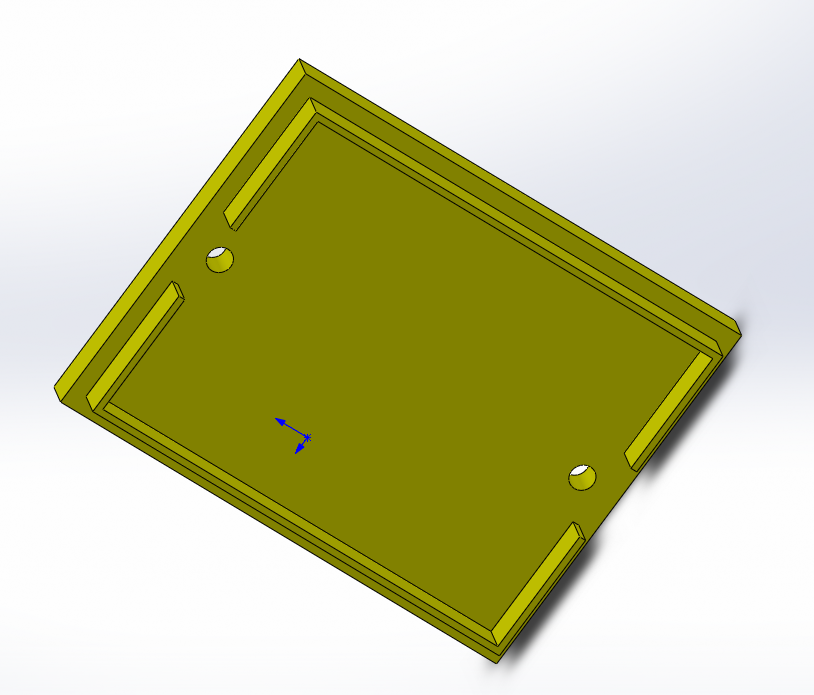
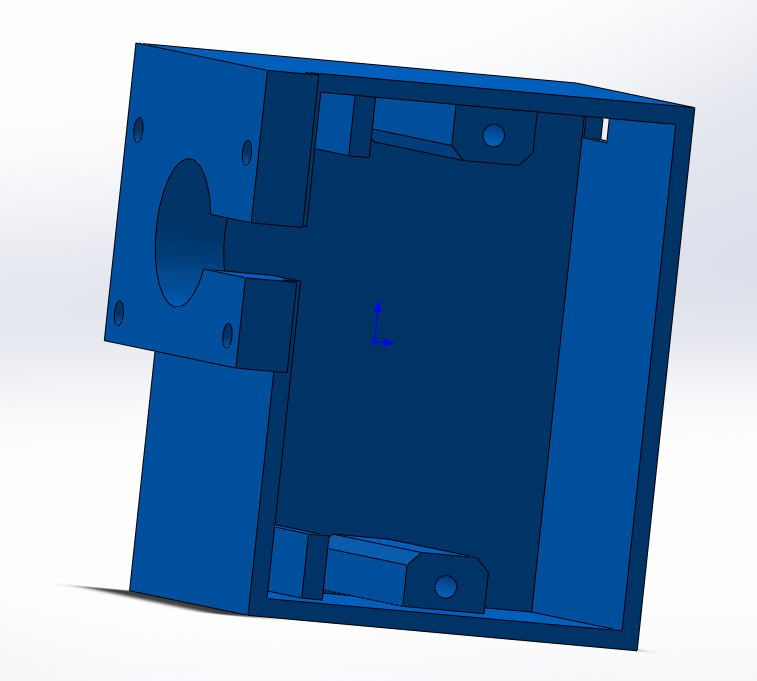
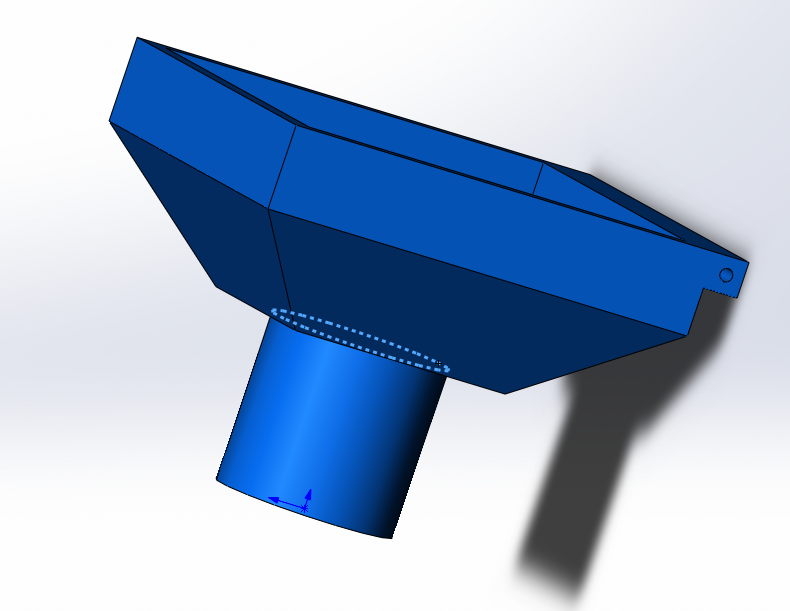
# Принцип роботи

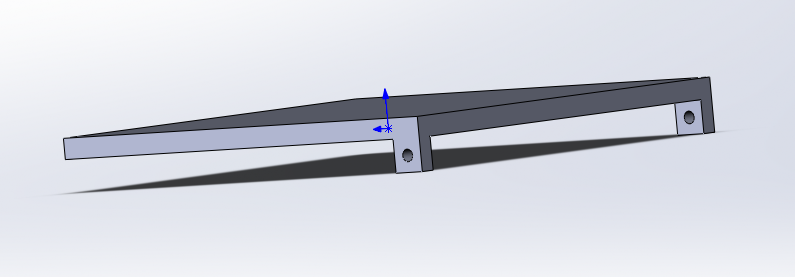
Всі деталі корпусу та кріплення, які використовувалися у проєкті, були створенні у програмі **SolidWorks 2019**, та надруковані на 3D-принтері.

**Електрогодівничка**

****

Електрогодівника складається з 6 деталей:

1. Шнек
2. Трубка-з’єднання
3. Коробки для двигуна
4. Кришки для двигуна
5. Кормовий бочок
6. Кришки для бочка



# ВИСНОВОК

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

У ході виконання даної наукової роботи був розроблений власний програмний комплекс, що надає користувачу інформацію про стан акваріумного середовища на свій смартфон у реальному часі, та має зрозумілий інтерфейс. За допомогою веб-інтерфейсу можна запланувати сонячний день в акваріумі, а також розклад годування рибок.

Розроблений програмний продукт має перспективи для використання у навчальних цілях, також для власних досліджень. Використана програмна частина дозволяє підключити різноманітні датчики, що розширює межі використання програмного комплексу.

У майбутньому планується розмістити та підключити камеру на 2-мегапікселя, яка працює з контролером ESP32, для дистанційного перегляду акваріуму в будь-який час.

# Список використаних джерел

Книжечки\*

* <https://www.youtube.com/watch?v=Lw-AT77I3GU>
* <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>
* <https://www.youtube.com/channel/UC4fc5wHqEoY3Ro3mu2IUOew>
* <https://www.youtube.com/watch?v=dWM4p_KaTHY&ab_channel=RuiSantos>
* <https://mounishkokkula.wordpress.com/how-the-5v-relay-works/>
* <https://materializecss.com/color.html>
* <https://ru.stackoverflow.com/>
* <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>