

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

Наукові керівники:

Ольховик Вадим Леонідович,  
керівник гуртка 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





Місце  
для фото

## 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  
Науковий керівник:  
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 пристроїв, що здатні керувати параметрами середовища в акваріумі через Інтернет, а також цілодобово їх моніторити.

Досліджено саму концепцію смарт акваріума, зроблено огляд існуючих комерційних розумних пристроїв для годування рибок в акваріумі, та онлайн моніторингу стану водного середовища.

На основі проведеного аналізу існуючих зразків, було спроектовано власний пристрій, та виготовлено робочий макет за допомогою 3D принтера.

Програмна частина створена на платформі Arduino, та включає в себе веб сервер, та користувацький інтерфейс 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

Вступ .....	5
РОЗДІЛ 1. Що таке Смарт-акваріум і як він працює? .....	7
РОЗДІЛ 2. Подібні проєкти.....	9
РОЗДІЛ 3. Використані модулі та їх роль у проєкті.....	11
РОЗДІЛ 4. Програмний код .....	13
РОЗДІЛ 5. 3D моделі .....	20
Висновок.....	24
Список використаних джерел.....	25

# 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 смартфона 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 - спроектувати “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.

Проект «Смарт-акваріум» має у собі відразу декілька корисних функцій: 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-стрічці, 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 Xiaomi зі своїми **Fish Tank Pro** «рис 2.1» і **YouPin** «рис 2.2 », компанія AqueE1 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 **Noison** «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 2.3 SHRIMP SET SMART



Рис 2.2 YouPin



Рис 2.4 Noison



**Рис 2.5 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. Двигун постійного струму (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).

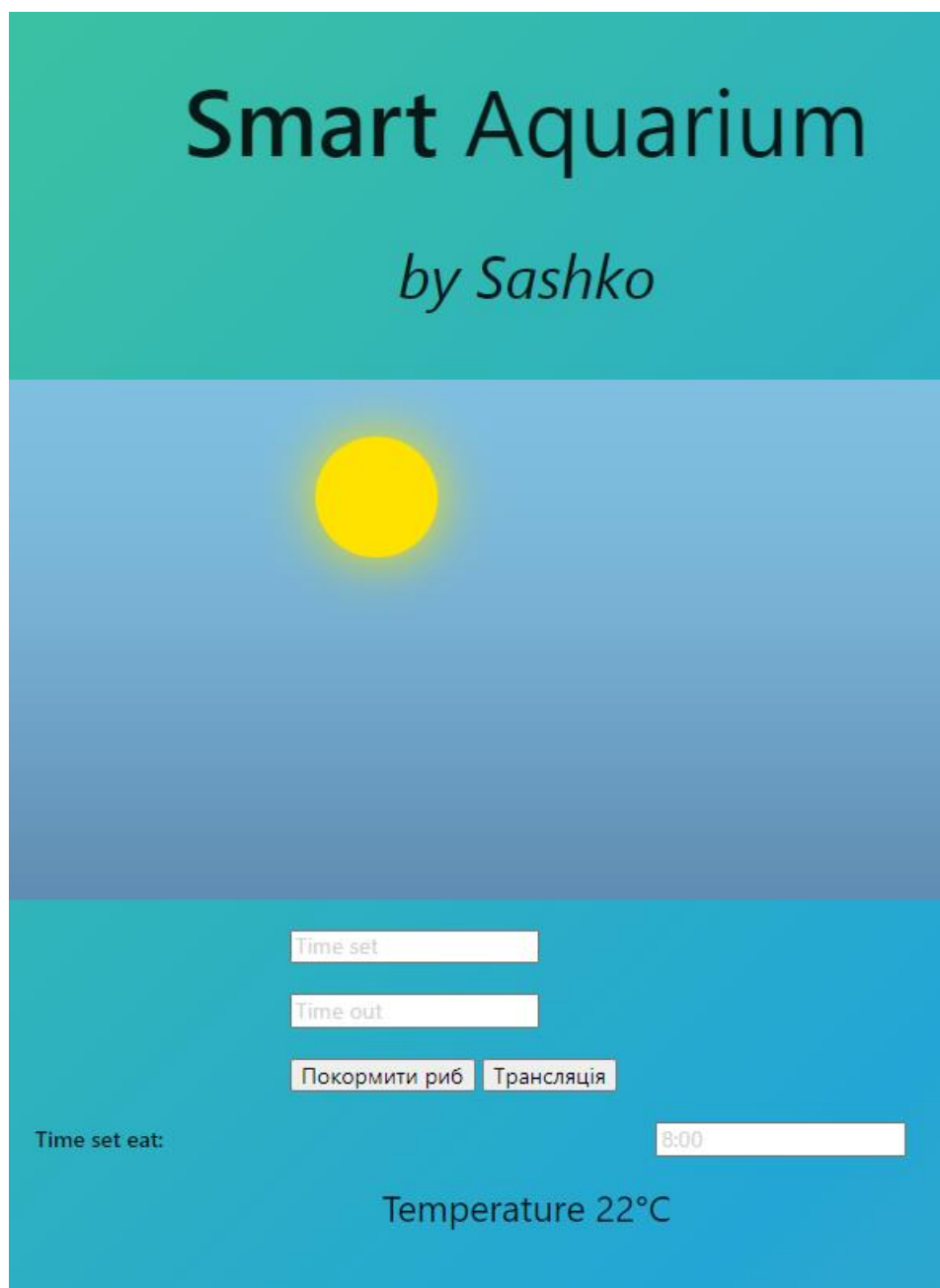
Також хотілося б звернути увагу найголовнішу частину цього проєкту - контролер. Чому вибір припав саме на ESP32? Чому ми не обрали інший контролер?

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 ESP32, маючи

невеликі розміри та оптимальну ціну, стає прекрасним вибором для заявленого проєкту.

## РОЗДІЛ 4. Програмний код

Однією з найважливіших цілей у проєкті є створення вебсайту для дистанційного керування. На сьогодні сайт виглядає наступним чином:



На сайті Ви можете вказати такі параметри: 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 трансляції 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();
    if (client && client_flag) {

        client_flag = false;
        currentTime = millis();
        previousTime = currentTime;
        Serial.println("New Client.");
        String currentLine = "";
        while (client.connected() && currentTime - previousTime <= timeoutTime ) {
            //currentTime = millis();
            if (client.available()) {
                char c = client.read();
                Serial.write(c);
                header += c;
                if (c == '\n') {
                    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");

}

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+"°C</h5> </div></div></body></html>");

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

delay(5);

break;

}

else {

    currentLine = "";

}

}

else if (c != '\r') {,

    currentLine += c;

}

```

```

    }

}

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);

}

}

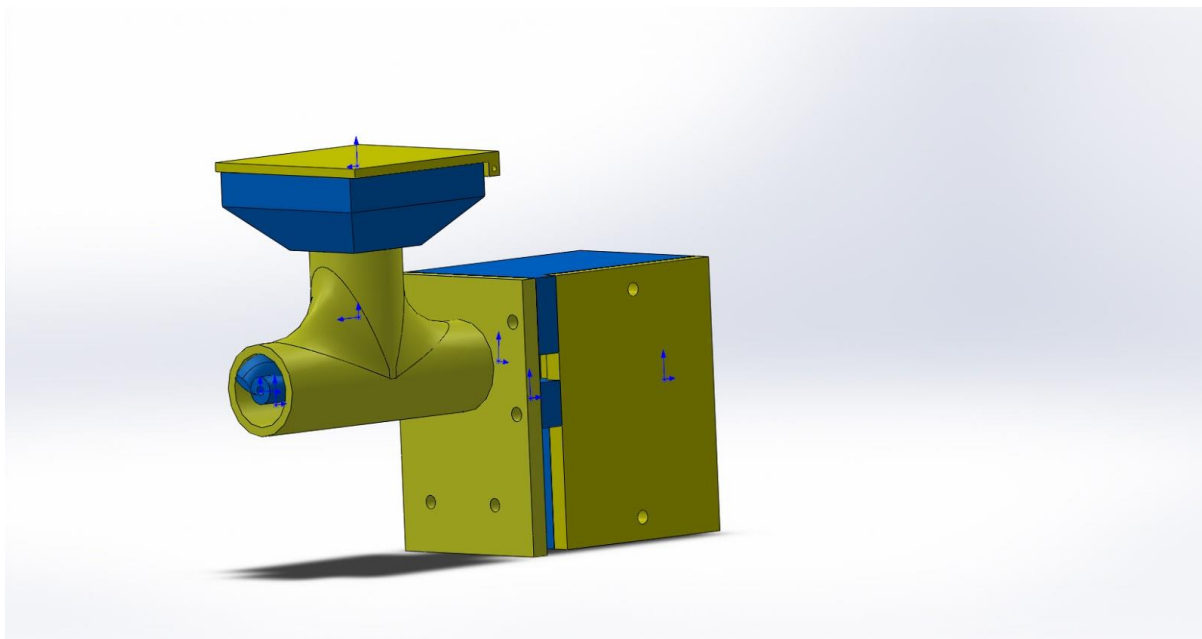
```

## РОЗДІЛ 5. 3D моделі

### Принцип роботи

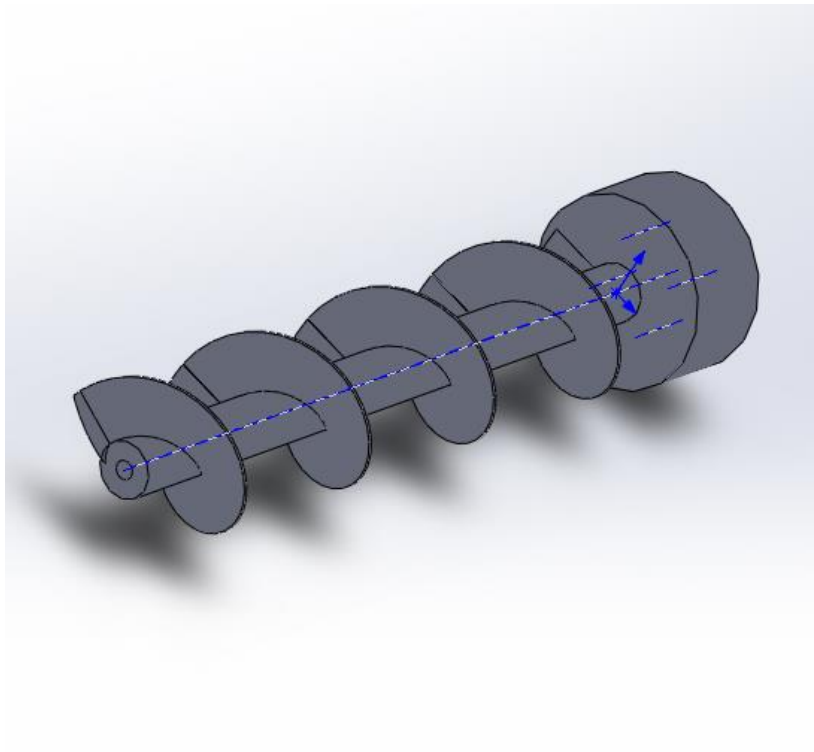
Всі деталі корпусу та кріплення, які використовувалися у проєкті, були створенні у програмі **SolidWorks 2019** та надруковані на 3D-принтері.

#### Електрогодівничка

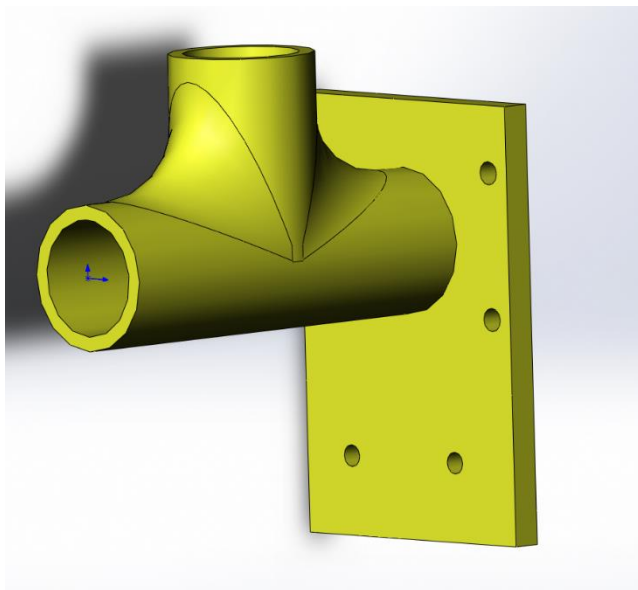


Електрогодівника складається з 6 деталей:

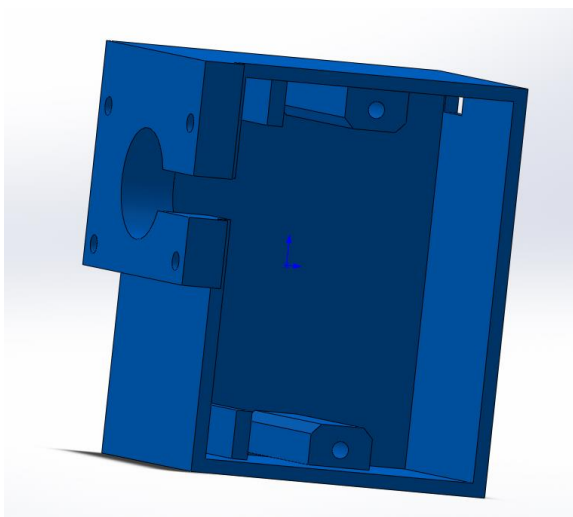
### 1. Шнек



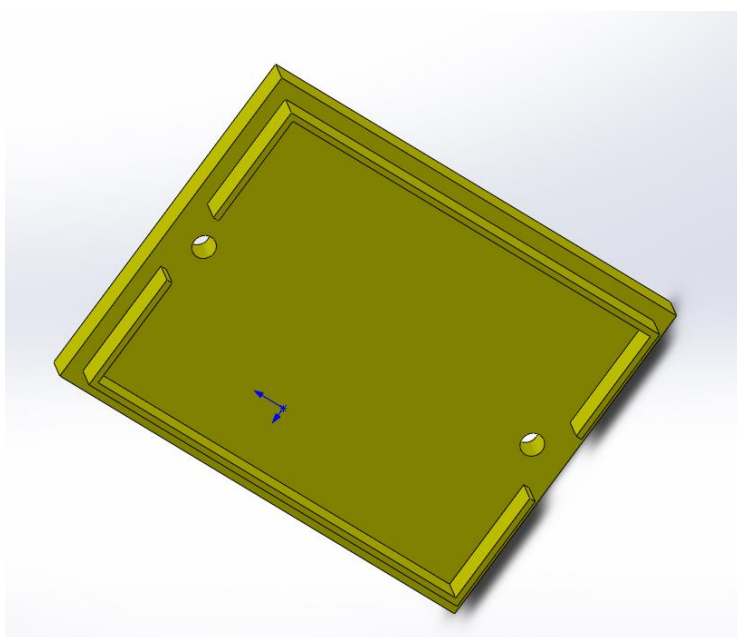
### 2. Трубка-з'єднання



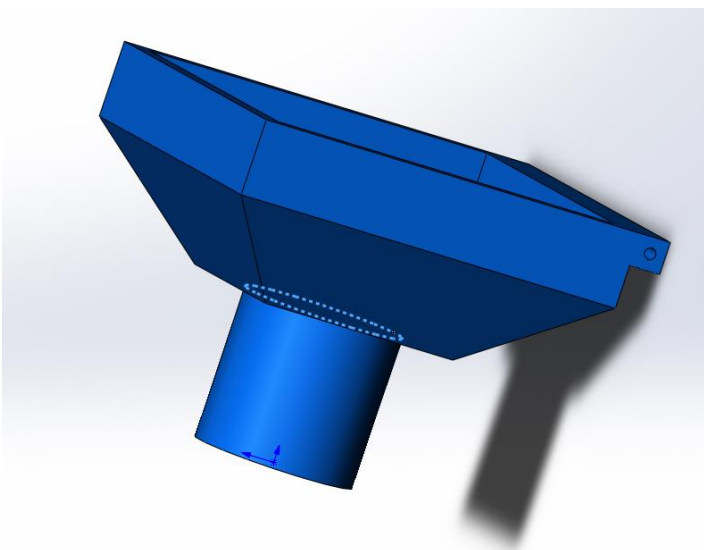
### 3. Коробки для двигуна



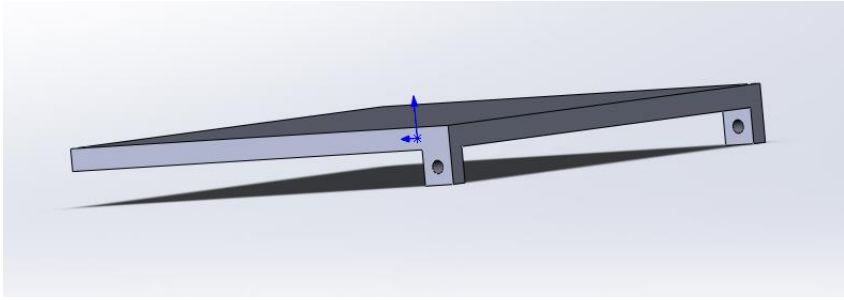
4. Кришки для двигуна



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 вказаної наукової роботи був розроблений власний програмний комплекс, що надає користувачу інформацію про стан акваріумного середовища на смартфон у реальному часі, має зрозумілий інтерфейс. За допомогою вебінтерфейсу 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 дозволяє розширити межі використання програмного комплексу.

У майбутньому планується розмістити камеру на 2-мегапікселя, яка вбудована у контролер ESP32, для дистанційного перегляду акваріума в будь-який зручний час.



# Список використаних джерел

- Программирование для детей. От основ к созданию роботов. Воронин И., Воронина В. Питер
- Изучаем Arduino: инструменты и методы технического волшебства Джереми Блум
- Мобильные роботы на базе ESP32 Момот Михаил Викторович
- IoT Development for ESP32 and ESP8266 with JavaScript: A Practic Guide to XS and the Moddable SDK 1st ed. Edition Peter Hoddie
- <https://www.youtube.com/watch?v=Lw-AT77I3GU>
- [https://www.youtube.com/watch?v=S0tb3jnf4w4&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/watch?v=S0tb3jnf4w4&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://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](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)