



Федерация Спортивной и Образовательной
робототехники

Российская Робототехническая Олимпиада 2024
Творческая категория
«Роботы и роботизированные системы в нефтегазовой отрасли»

ОТЧЕТ по проекту «УМНЫЙ СКЛАД» ДЛЯ РЕГИСТРАЦИИ

Команда «Новаторов»

Младшая возрастная категория

Выполнили:

Дмитрий Коннов, ученик 5 класса
МАОУ лицей № 121,
Александр Романов, ученик 5
класса МБОУ СОШ лицей № 5.

Руководитель:

Харисова Земфира Рашидовна
наставник Клуба робототехники и
детского творчества «MakerLab»

г. Казань,

2024



Содержание

Аннотация.....	2
О команде.....	3
Введение.....	4
Глава 1 Конструирование прототипа.....	5
Глава 2 Датчики в системах прототипа.....	6
Глава 3 Программа	7
Глава 4 Экономическая часть.....	10
Глава 5 Эксперимент.....	11
Заключение.....	13
Развитие проекта.....	15
Итоги.....	17
Список литературы и источники информации	19
Приложения	20



Отчет для РЕГИСТРАЦИИ. Отчет на оценку будет загружен позже.

АННОТАЦИЯ



О КОМАНДЕ

Наша команда из г. Казани Республики Татарстан. Мы представляем проект «Умный склад». Мы занимаемся в клубе робототехники и детского творчества «MakerLab». Наша команда «Новаторы» (рис. 1):

Дмитрий Коннов (ученик 5 класса, МАОУ лицей №121) – капитан команды и программист на языке Python, проектировщик в программе Компас 3D.

Александр Романов (ученик 5 класса, МБОУ СОШ лицей № 5) - инженер-разработчик, инженер-конструктор и программист в Arduino IDE на языке C.

Мы занимаемся робототехникой 6 лет. Имеем несколько побед в мероприятиях республиканского и федерального уровня. На Республиканском этапе Робототехнической олимпиады мы успешно выступили и заняли 1 место. Тренер Земфира Харисова. Девиз нашего проекта «Скажем ДА умному складу, скажем НЕТ дронам и авариям!!!».



Рис. 1 Наша команда (Дима, Саша)

ВВЕДЕНИЕ

Мы решили выбрать тему задания «Разработка роботов и роботизированных систем для создания и улучшения складских помещений в нефтегазовой отрасли». Потому что, склады нефтепродуктов являются опасными объектами и нефтепродукты токсичны, способны сами воспламениться и сильно гореть. К таким объектам предъявляются повышенные требования, есть специальный СНИП 2.11.03-93 "Склады нефти и нефтепродуктов. Противопожарные нормы".

Сегодня в России насчитывается около 120 нефтеперерабатывающих заводов. У каждого крупного предприятия существуют большое количество складов, которые до сих пор работают в ручном или полуавтоматическом режиме, что сильно снижает их эффективность. Ошибки людей на складах, могут привести к экологическим авариям (загрязнению окружающей среды). Фото аварии (рис. 2):



Рис. 2 Фото аварий на складах

Например, в 2023 году в Енисейском районе произошел разлив дизельного топлива на складе временного хранения ГСМ, вылились 15 тонн нефтепродуктов. Причина разлива - неисправность запорной арматуры, которая была установлена на емкости для хранения дизельного топлива. В районе ввели режим чрезвычайной ситуации.

Основные причины аварий на складах и наливных станциях: человеческий фактор, износ оборудования и новые причины – это атака нефтяных объектов беспилотными летательными аппаратами (БПЛА). Только в апреле-мае 2024 года были множественные атаки дронов на нефтеперерабатывающие предприятия Татарстана (рис. 3).



Рис. 3 Фото нападения дронов и аварии на нефтяных предприятиях



Проблема: обеспечение безопасности людей, работающих на взрывоопасных складах.

Цель работы: уменьшить количество аварий на складах нефтепродуктов путем создания полностью роботизированного безопасного склада нефтепродуктов.

Аналоги умного склада - водители на погрузчиках, автономные грузчики, большой умный склад от компании ГАЗПРОМ (но там все равно есть люди) и подобные. Сейчас на складах отсутствует система обнаружения и оповещения дронов.

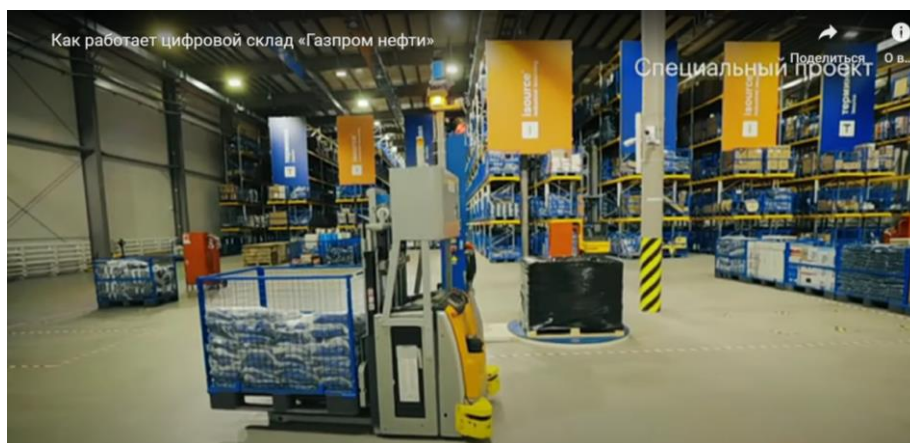


Рис. 4 Фото работы цифрового склада компании ГАЗПРОМ

Существуют различные системы и комплексы для обнаружения и борьбы с беспилотными летательными аппаратами (БПЛА) (рис. 5). Например, экосистема Kaspersky Antidrone позволяет использовать широкий спектр сенсоров для сканирования воздушного пространства и обнаружения беспилотных летательных аппаратов (БПЛА). Устройства интегрируются в комплекс в соответствии с конкретными потребностями и моделью угроз.



Рис. 5 Пример фото систем и комплексов для обнаружения и подавления

Главные методы обнаружения БПЛА включают акустический, оптический, тепловой, радиочастотный и радарный. Каждый из них имеет свои особенности и области применения. Сейчас на складах предприятий нефтегазовой отрасли не применяются системы и комплексы обнаружения и борьбы с БПЛА.

Мы с командой начали работу в январе. Распределили роли в команде. Этапы нашей разработки проекта: исследование предметной области, моделирование, создание прототипа, тестирование, корректировка.

Отчет для РЕГИСТРАЦИИ. Отчет на оценку будет загружен позже.

Наше решение – полностью роботизированный безопасный склад нефтепродуктов.

Комплект включает в себя несколько систем: 1. Система хранения, транспортировки и налива нефтепродуктов. 2. Система обнаружения дронов и оповещения персонала. В дальнейшем предполагается реализация системы уничтожения беспилотных летательных аппаратов.

Технологии, используемые в проекте – компьютерное зрение, проектирование, 3-д моделирование и 3-д печать, интернет вещей.

На роботизированных складах, включающих систему налива, все делают роботы, а люди забирают бочки с нефтепродуктами на машинах, не выходя из них. Предлагаемая система отслеживания дронов позволит своевременно оповестить персонал, например водителей автомобилей, технический персонал склада и других людей, которые могут оказаться в зоне поражения.

Ссылки на источники в списке литературы.



ГЛАВА 1 КОНСТРУИРОВАНИЕ ПРОТОТИПА

Идея проекта «УМНЫЙ СКЛАД» пришла нам почти сразу, когда мы генерировали и предлагали идеи для предприятий нефтегазовой отрасли. Про обнаружение дронов мы добавили позже, когда дроны начали нападать на предприятия нашей республики. Для реализации идеи мы распределили роли: инженер-программист – занимается системой компьютерного зрения, изучает и внедряет алгоритмы, которые позволяют обнаруживать дроны; инженер-разработчик – занимается разработкой роботизированного решения, реализует систему управления роботом; инженер-проектировщик – занимается проектированием робота и системы налива, реализует 3-д печать элементов. Роль инженера-проектировщика поделили между собой.

Наше решение представлено несколькими системами: система хранения, транспортировки и налива нефтепродуктов; система обнаружения дронов и оповещения персонала.

На начальном этапе моделирования и прототипирования для системы транспортировки мы собрали из фанеры и пластика модель робо-грузчика. Всего таких роботов нужно 2. Также мы спроектировали станцию обнаружения дронов, которая стоит рядом со складом. Всего таких станций нужно 4. Спроектировали элементы для системы налива.

После тестирования прототипа из фанеры мы скорректировали нашу 3-д модель с учтенными недостатками в программе Компас 3D. Всего было спроектировано в программе Компас 3D семь элементов: для робо-грузчика – стенки, основание робота, крепления, корпус; для станции оповещения – опорные конструкции, для наливной станции – крыша налива с баком. Часть спроектированные элементы мы распечатали из пластика на 3-д принтере и часть вырезали на лазерном станке (стенки, основание, крепления и другие элементы) (рис. 6-11).

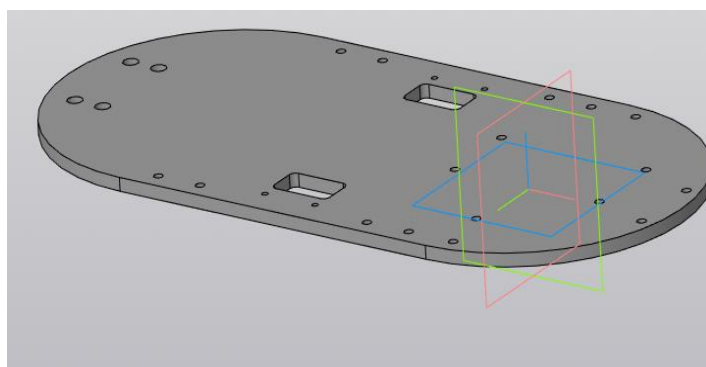


Рис. 6 Основание (1 этаж робота)

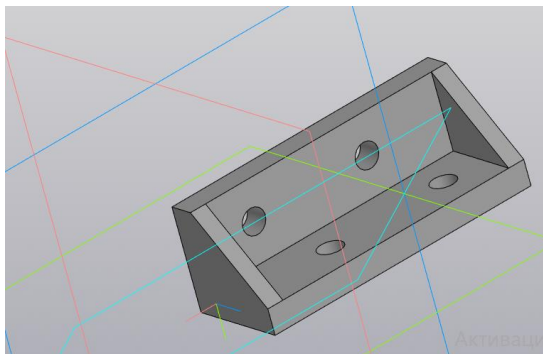


Рис. 7 Уголок для скрепления частей робота

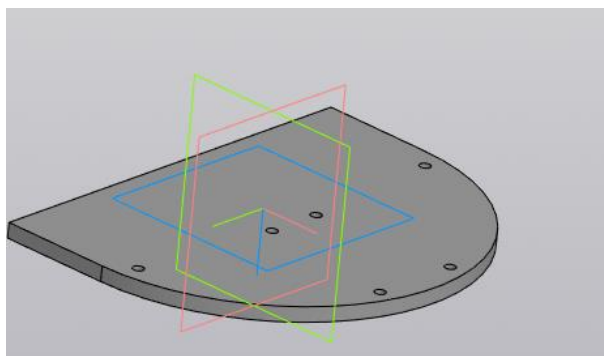


Рис. 8 Элемент корпуса (2 этаж робота)

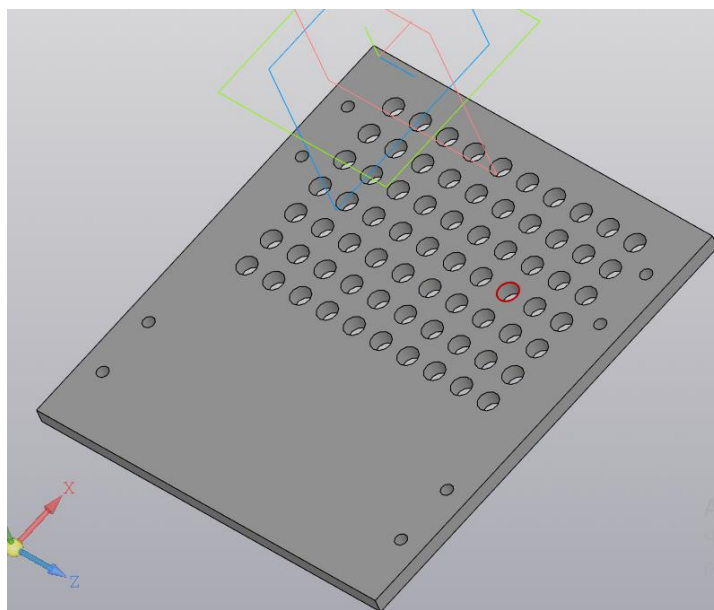


Рис. 9 Стенка робота

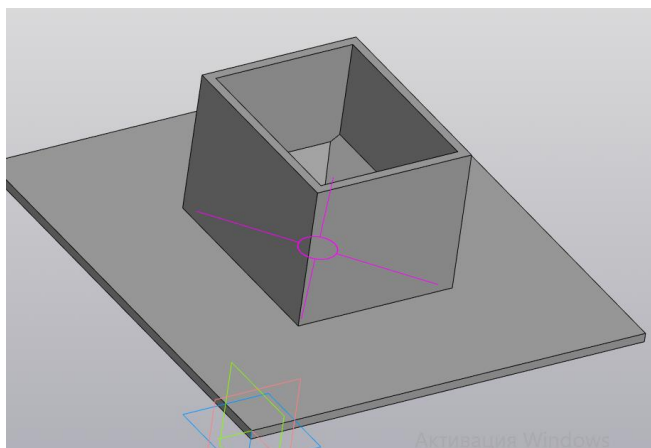


Рис. 10 Крыша налива с баком

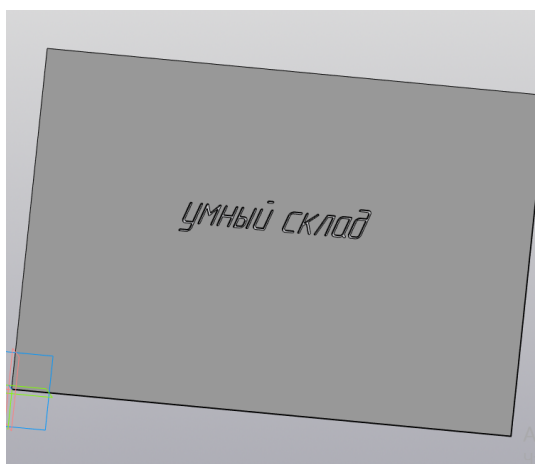
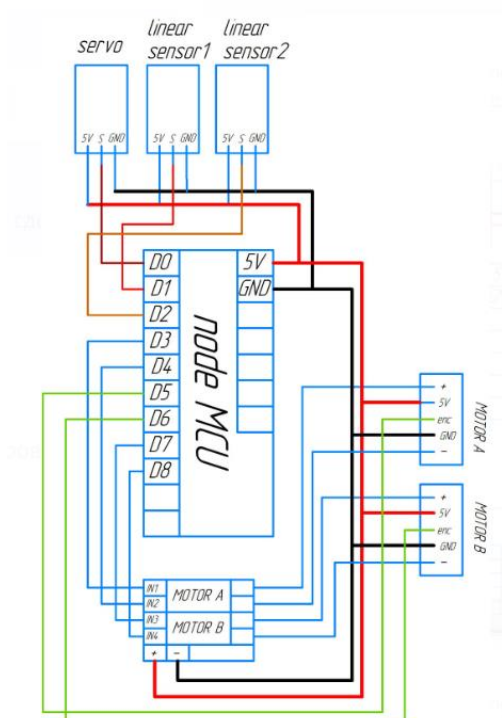


Рис. 11 Стенка налива

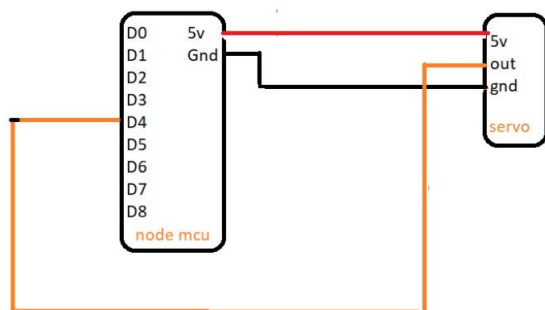
Техническая реализация Робот: плата NodeMcu v3 Lolin построена на микроконтроллере ESP8266 с интерфейсом Wi-Fi на базе микросхемы CH340G, драйвер МХ1508, аккумуляторный блок, моторы n20, датчики линии TCRT5000, сервоприводы, камера и др.



Функции:

Перемещение бочек по складу и наполнение с помощью станции налива

Техническая реализация Налив: плата NodeMcu v3 Lolin построена на микроконтроллере ESP8266 с интерфейсом Wi-Fi на базе микросхемы CH340G, servo мото



Функции:

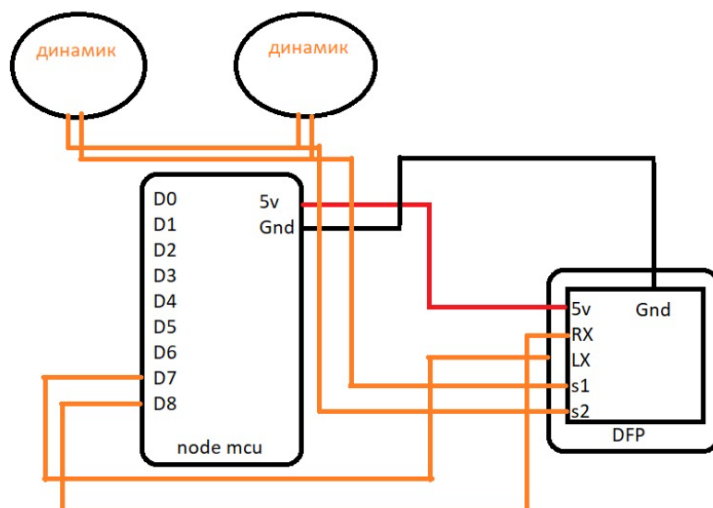
Залив нефтепродукта в бочку .Когда подъезжает робот

Техническая реализация Телебудка

Робот: плата NodeMcu v3 Lolin построена на микроконтроллере ESP8266 с интерфейсом Wi-Fi на базе микросхемы CH340G, расширительной платы nodemcu base, плата DFPlacer , 2 динамика

Функции:

Определение и оповещение о дроне



Материалы:

- 1) Программы - Язык программирования Python (распознавания образа дрона), Arduino IDE на языке C (движение робота и налив), Компас 3D (построение модели)
- 2)
- 3) Создание модели - лазерный станок, 3д принтер и др.

Материалы:

- 2) Техническая реализация – плата NodeMcu v3 Lolin построена на микроконтроллере ESP8266 с интерфейсом Wi-Fi на базе микросхемы CH340G, драйвер MX1508, аккумуляторный блок, моторы n20, датчики линии TCRT5000, сервоприводы, камера и др.
- 3) Создание модели - лазерный станок, 3д принтер и др.

Две системы и фото.

Краткое описание аналога в реале и как в модели.



ИТОГИ

Роботизация позволит повысить эффективность работы склада и снизить число аварий. Мы за безопасный, умный, эффективный склад!!!

Отчет, видеоролик и дополнительную информацию по нашему проекту «УМНЫЙ СКЛАД» можно посмотреть в папке по ссылке <https://cloud.mail.ru/public/64CS/fN2FXhJLW>

СПИСОК ЛИТЕРАТУРЫ И ИСТОЧНИКИ ИНФОРМАЦИИ

1. Статистика чрезвычайных происшествий на объектах нефтеперерабатывающей и нефтехимической промышленности за 2007-2016 гг. <https://ogbus.ru/article/view/statistika-chrezvychajnyx-proisshestvij-na-obektax-nefteperera>
2. Очевидцы выкладывают кадры атаки беспилотника в Татарстане https://dzen.ru/video/watch/660b9a5d2619a736fb04b467?share_to=whatsapp
3. Нападение дронов <https://yandex.ru/video/touch/preview/18019208427806751216>
4. Примеры систем распознавания <https://sky-x.pro/blog/opticheskiy-kontroly--obnaruzhenie-bpla-chasty-5>
5. Примеры систем распознавания <https://antidrone.kaspersky.com/ru/solution/hardware>
6. Система компании Газпром <https://www.tek-all.ru/news/id8283-gazprom-neft-pokazala-umnie-skladi-v-hmao/>



ПРИЛОЖЕНИЕ 1. Программа для системы обнаружения дронов и оповещения сотрудников склада

```
import cv2
import math
import numpy
from paho.mqtt.subscribe import simple
from paho.mqtt.publish import single
import paho.mqtt.client as m
import time
def fmar(conturs,image):
    rect = cv2.minAreaRect(conturs)
    x,y = rect[0]
    cv2.circle(image,(int(x),int(y)),6,(0,0,255),-1)
    points = cv2.boxPoints(rect)
    point_int = numpy.array(points,int)
    cv2.drawContours(image,[point_int],-1,(255,0,0),3)
    return x,y

def read_values_from_file(filename):
    file = open(filename, 'r')
    values = file.read()
    values_split = values.split(',')
    for number in range(6):
        values_split[number] = int(values_split[number])
    file.close()
    return values_split
h = 0
cap = cv2.VideoCapture(0)
hostname = 'm9.wqtt.ru'
user_name = "u_JA38QM"
password = "XQAQsv4b"
port = 16922
topik1 = 'user_57fda11f/SIRENA1'
my_client = m.Client()
my_client.username_pw_set(user_name, password)
my_client.connect(hostname, port)
while True:
    isRead, image = cap.read()

    h_up_car, h_down_car, s_up_car, s_down_car, v_up_car, v_down_car =
read_values_from_file('red_hsv.txt')

    HSV_up_car = [h_up_car, s_up_car, v_up_car]
```



Отчет для РЕГИСТРАЦИИ. Отчет на оценку будет загружен позже.

```
HSV_down_car = [h_down_car, s_down_car, v_down_car]

np_HSV_up_car = numpy.array(HSV_up_car)
np_HSV_down_car = numpy.array(HSV_down_car)

image_hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV_FULL)

mask = cv2.inRange(image_hsv, np_HSV_down_car, np_HSV_up_car)

contours,cyrvis
cv2.findContours(mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_NONE)

for contours in contours:
    points_cont = len(contours)
    if points_cont > 50 and h==0:
        fmar(contours, image)
        my_client.publish(topik1,2)
cv2.imshow('window', image)
key = cv2.waitKey(20)
cap.release()
```



ПРИЛОЖЕНИЕ 2. Программа для системы хранения и транспортировки нефтепродуктов

```
#include <ESP8266WiFi.h> //подключение библиотеки
#include <PubSubClient.h> //подключение библиотеки для работы протокола MQTT
#include <Servo.h> //подключение библиотек для работы серво привода
#define IN1 D3 //левый
#define IN2 D4 //левый
#define IN3 D7 //правый
#define IN4 D8 //правый
#define Ser D0
Servo myservo;
WiFiClient esp;
PubSubClient board(esp); //создание объекта board
const int pin_SENSOR_r = D2;
const int pin_SENSOR_l = D1;
const byte l = D5; //введем константу пина прерывания
const byte r = D6; //введем константу пина прерывания
int enc_r = 0; //переменные для хранения значения правого датчика, нач. знач.=0
int enc_l = 0; //переменные для хранения значения правого датчика, нач. знач.=0
int sensorState_l = 0;
int sensorState_r = 0;
char payload_message[3]; //создадим массив для принимаемого значения
int flag = 0;
int power = 0;
int table = 0;
int shim = 165;
int i = 0;
int forward_angle = 60;
void ICACHE_RAM_ATTR encoder_right() { //функции привязанные к срабатыванию
датчика////////////////////////////////////
enc_r++; //действие по увеличению переменной
}
void ICACHE_RAM_ATTR encoder_left() { //функции привязанные к срабатыванию
датчика////////////////////////////////////
enc_l++; //действие по увеличению переменной
}

void receive_message(char* topic, byte* payload, int length) {
int table = 0;
int i = 0;
for (int j = 0; j < length; j++) {
payload_message[j] = payload[j];
}
```



Отчет для РЕГИСТРАЦИИ. Отчет на оценку будет загружен позже.

```
    }  
    // table = atoi(payload_message);  
    Serial.println(payload_message);  
    board.publish("aliPrimo888", "stop");    //и отправим на брокер сообщение о  
подключении  
}
```

```
void stop_hold() {  
    analogWrite(IN1, shim);  
    analogWrite(IN2, shim);  
    analogWrite(IN3, shim);  
    analogWrite(IN4, shim);  
    delay(100);  
    analogWrite(IN1, 0);  
    analogWrite(IN2, 0);  
    analogWrite(IN3, 0);  
    analogWrite(IN4, 0);  
    delay(500);  
}
```

```
void ezda_po_lineyke(int perek) {  
    i = 0;  
    while (i < perek) {  
        sensorState_l = digitalRead(pin_SENSOR_l);  
        sensorState_r = digitalRead(pin_SENSOR_r);  
  
        if (sensorState_l == 0 && sensorState_r == 1) {  
            analogWrite(IN1, shim); //левый вперед  
            analogWrite(IN2, 0);  
            analogWrite(IN3, shim - 60); //правый вперед  
            analogWrite(IN4, 0);  
            delay(180);  
        } else if (sensorState_r == 0 && sensorState_l == 1) {  
            analogWrite(IN1, shim - 60); //правый вперед  
            analogWrite(IN2, 0);  
            analogWrite(IN3, shim);  
            analogWrite(IN4, 0);  
            delay(180);  
        } else if (sensorState_r == 1 && sensorState_l == 1) {  
            analogWrite(IN1, shim); //правый вперед  
            analogWrite(IN2, 0);  
            analogWrite(IN3, shim);  
            analogWrite(IN4, 0);  
            delay(180);  
        }  
        i++;  
    }  
}
```



```
} else if (sensorState_r == 0 && sensorState_l == 0) {
  /*
  analogWrite(IN1, 0); //правый вперед
  analogWrite(IN2, 0);
  analogWrite(IN3, 0);
  analogWrite(IN4, 0);
  delay(1000);
  */
  stop_hold();
  forward_align(shim, forward_angle);
  i++;
}
}
stop_hold();
}

void ezda_po_lineyke_without_forward() {
  i = 0;
  while (i < 1) {
    sensorState_l = digitalRead(pin_SENSOR_l);
    sensorState_r = digitalRead(pin_SENSOR_r);

    if (sensorState_l == 0 && sensorState_r == 1) {
      analogWrite(IN1, shim); //левый вперед
      analogWrite(IN2, 0);
      analogWrite(IN3, shim - 60); //правый вперед
      analogWrite(IN4, 0);
      delay(180);
    } else if (sensorState_r == 0 && sensorState_l == 1) {
      analogWrite(IN1, shim - 60); //правый вперед
      analogWrite(IN2, 0);
      analogWrite(IN3, shim);
      analogWrite(IN4, 0);
      delay(180);
    } else if (sensorState_r == 1 && sensorState_l == 1) {
      analogWrite(IN1, shim); //правый вперед
      analogWrite(IN2, 0);
      analogWrite(IN3, shim);
      analogWrite(IN4, 0);
      delay(180);
    } else if (sensorState_r == 0 && sensorState_l == 0) {
      analogWrite(IN1, 0); //правый вперед
      analogWrite(IN2, 0);
      analogWrite(IN3, 0);
    }
  }
}
```



Отчет для РЕГИСТРАЦИИ. Отчет на оценку будет загружен позже.

```
    analogWrite(IN4, 0);
    delay(1000);
    i++;
  }
}

void turn_left_datchik_right() {
  delay(100);
  sensorState_r = digitalRead(pin_SENSOR_r);
  delay(100);
  sensorState_r = digitalRead(pin_SENSOR_r);
  sensorState_r = digitalRead(pin_SENSOR_r);
  while (sensorState_r == 1) {
    sensorState_r = digitalRead(pin_SENSOR_r);
    analogWrite(IN1, shim - 30);
    analogWrite(IN2, 0);
    analogWrite(IN3, 0);
    analogWrite(IN4, shim - 30);
  }
  stop_hold();
}

void turn_right_datchik_left() {
  sensorState_l = digitalRead(pin_SENSOR_l);
  while (sensorState_l == 1) {
    sensorState_l = digitalRead(pin_SENSOR_l);
    analogWrite(IN1, 0);
    analogWrite(IN2, shim - 30);
    analogWrite(IN3, shim - 30);
    analogWrite(IN4, 0);
  }
  stop_hold();
}

void left_align(int power, int angle) {
  int power_l = power;
  int power_r = power;
  int delta = 0;
  enc_r = 0;
  enc_l = 0;
  while (enc_r < angle * 7.5) {
    delta = (enc_l - enc_r) / 2;
    analogWrite(IN1, power_l);
```



```
    analogWrite(IN2, 0);
    analogWrite(IN3, 0);
    analogWrite(IN4, power_r);
    Serial.print(enc_l);
    Serial.print(" ");
    Serial.println(enc_r);
}
stop_hold();
}

void back(int power, int turnover) {
    int power_l = power + 15;
    int power_r = power;
    int delta = 0;
    enc_r = 0;
    enc_l = 0;
    while (enc_l < turnover * 10) {
        delta = (enc_l - enc_r) / 8;
        analogWrite(IN1, 0);
        analogWrite(IN2, power_l);
        analogWrite(IN3, 0);
        analogWrite(IN4, power_r);
        // Serial.print(enc_l);
        // Serial.print(" ");
        // Serial.println(enc_r);
    }
}

void right_align(int power, int angle) {
    int power_l = power;
    int power_r = power;
    int delta = 0;
    enc_r = 0;
    enc_l = 0;
    while (enc_r < angle * 7.2) {
        delta = (enc_l - enc_r) / 2;
        analogWrite(IN1, 0);
        analogWrite(IN2, power_l);
        analogWrite(IN3, power_r);
        analogWrite(IN4, 0);
        Serial.print(enc_l);
        Serial.print(" ");
        Serial.println(enc_r);
    }
}
```




```
    stop_hold();
}

void forward_align(int power, int turnover) {
    int power_l = power;
    int power_r = power;
    int delta = 0;
    enc_r = 0;
    enc_l = 0;
    while (enc_l < turnover * 10) {
        delta = (enc_l - enc_r) / 8;
        analogWrite(IN1, power_l);
        analogWrite(IN2, 0);
        analogWrite(IN3, power_r);
        analogWrite(IN4, 0);
        // Serial.print(enc_l);
        // Serial.print(" ");
        // Serial.println(enc_r);
    }
    stop_hold();
}

void servo() {
    // myservo.write(0);
    // delay(1000);
    // myservo.write(90);
    // delay(1000);
    myservo.write(180);
    delay(400);
    myservo.write(90);
    delay(1000);
}

void servo_CLOCK() {
    // myservo.write(0);
    // delay(1000);
    // myservo.write(90);
    // delay(1000);
    myservo.write(0);
    delay(400);
    myservo.write(90);
    delay(1000);
}
```



Отчет для РЕГИСТРАЦИИ. Отчет на оценку будет загружен позже.

```
void setup() {
  pinMode(pin_SENSOR_r, INPUT);
  pinMode(pin_SENSOR_1, INPUT);
  Serial.begin(9600);
  WiFi.begin("zemfira", "12345678"); // подключение платы к WIFI
  pinMode(LED_BUILTIN, OUTPUT); //настройка пина
  board.setServer("95.163.230.191", 1883); //установка адреса и порта
  board.setCallback(receive_message); //вызов функции при принятии сообщения
  attachInterrupt(1, encoder_left, RISING); //функция для настройки прерываний
  attachInterrupt(r, encoder_right, RISING); //функция для настройки прерываний
  myservo.attach(Ser);
}

void loop() {

  /*
  if (WiFi.status() == WL_CONNECTED && board.connected() != 0) //если плата
  подключена к WIFI и к брокеру
  {
    digitalWrite(LED_BUILTIN, LOW); // то включаем светодио
    if (flag == 0) {
      board.publish("aliPrimo888", "ready"); //и отправим на брокер сообщение о
  подключении
      flag = 1;
    }
  } else if (WiFi.status() == WL_CONNECTED && board.connected() == 0) //если плата
  подключена к WIFI, но не к брокеру
  {
    board.connect("esp_we12", "maker", "LABmaker123"); // то подключаемся к брокеру
    board.subscribe("aliPrimo999"); //подключимся к топику для принятия
  сообщения
    board.subscribe("naliv_2");
    digitalWrite(LED_BUILTIN, LOW); // и включаем светодиод
    delay(1000);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(1000);
  } else {
    digitalWrite(LED_BUILTIN, HIGH); // выключаем свтеодиод, если не подключены
  к WIFI
    Serial.println("no");
  }
  */
  ezda_po_lineyke(1);
  left_align(shim, 45);
}
```



```
turn_left_datchik_right();
ezda_po_lineyke_without_forward();
delay(1000);
servo();
left_align(shim, 80);
turn_left_datchik_right();
ezda_po_lineyke(1);
left_align(shim, 45);
turn_left_datchik_right();
ezda_po_lineyke(1);
ezda_po_lineyke(1);
servo_CLOCK();
forward_align(shim, 65);
// board.publish("naliv_2", "2");
delay(3000);
servo();
back(shim, 100);
left_align(shim, 130);
turn_left_datchik_right();
ezda_po_lineyke(1);
left_align(shim, 90);
turn_left_datchik_right();
ezda_po_lineyke(1);
servo_CLOCK();
// back(shim,60);
// left_align(shim, 190);
// forward_align(shim, 49);
// left_align(shim, 90);
// ezda_po_lineyke(1);
// forward_align(shim, 65);
// ezda_po_lineyke(1);
// left_align(shim, 90);
// ezda_po_lineyke(1);
// forward_align(shim, 65);
// ezda_po_lineyke(1);
// forward_align(shim, 65);

delay(10000);
}
```



ПРИЛОЖЕНИЕ 3. Программа для системы налива нефтепродуктов

```
#include <ESP8266WiFi.h> //подключение библиотеки
#include <PubSubClient.h> //подключение библиотеки для работы протокола MQTT
#include <Servo.h>
#define PIN_SERVO D4

WiFiClient esp;
PubSubClient board(esp); //создание объекта board
Servo myservo;
int flag = 0;

void receive_message(char* topic, byte* payload, int length) {
  myservo.write(0); // tell servo to go to position in variable 'pos'
  delay(3000);
  myservo.write(180); // tell servo to go to position in variable 'pos'
  delay(1000);
  board.publish("naliv", "stop"); //и отправим на брокер сообщение о подключении
}

void setup() {
  Serial.begin(9600);
  WiFi.begin("dima", "12345678"); // подключение платы к WIFI
  pinMode(LED_BUILTIN, OUTPUT); //настройка пина
  board.setServer("95.163.230.191", 1883); //установка адреса и порта
  board.setCallback(receive_message); //вызов функции при принятии сообщения
  myservo.attach(PIN_SERVO);
}

void loop() {
  if (WiFi.status() == WL_CONNECTED && board.connected() != 0) //если плата
  подключена к WIFI и к брокеру
  {
    digitalWrite(LED_BUILTIN, LOW); // то включаем светодио
    if (flag == 0) {
      board.publish("naliv", "ready"); //и отправим на брокер сообщение о подключении
      flag = 1;
    }
  } else if (WiFi.status() == WL_CONNECTED && board.connected() == 0) //если плата
  подключена к WIFI, но не к брокеру
  {
    board.connect("esp_we13", "maker", "LABmaker123"); // то подключаемся к брокеру
    board.subscribe("naliv_2"); //подключоимся к топику для принятия
    сообщения
  }
}
```



Отчет для РЕГИСТРАЦИИ. Отчет на оценку будет загружен позже.

```
digitalWrite(LED_BUILTIN, LOW);           // и включаем светодиод
delay(1000);
digitalWrite(LED_BUILTIN, HIGH);
delay(1000);
} else {
digitalWrite(LED_BUILTIN, HIGH); // выключаем светодиод, если не подключены
к WIFI
Serial.println("no");
}
board.loop();
}
```

