

**МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА №2»
ПРЕДГОРНОГО МУНИЦИПАЛЬНОГО ОКРУГА
СТАВРОПОЛЬСКОГО КРАЯ**

**Радиоуправляемая модель-платформа,
распечатанная на 3Dпринтере
«доставка пакета первой помощи»**

**Выполнил:
ученик 10 класса
МБОУ СОШ №2
Васильченко Тимофей**

**Руководитель:
Учитель труда
Юрченко О. В.**

**Предгорный округ
станция Суворовская 2024 год**

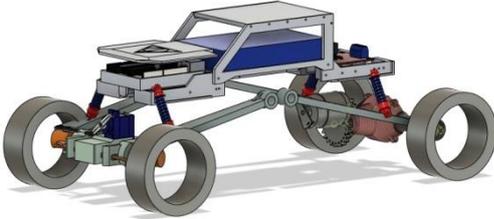
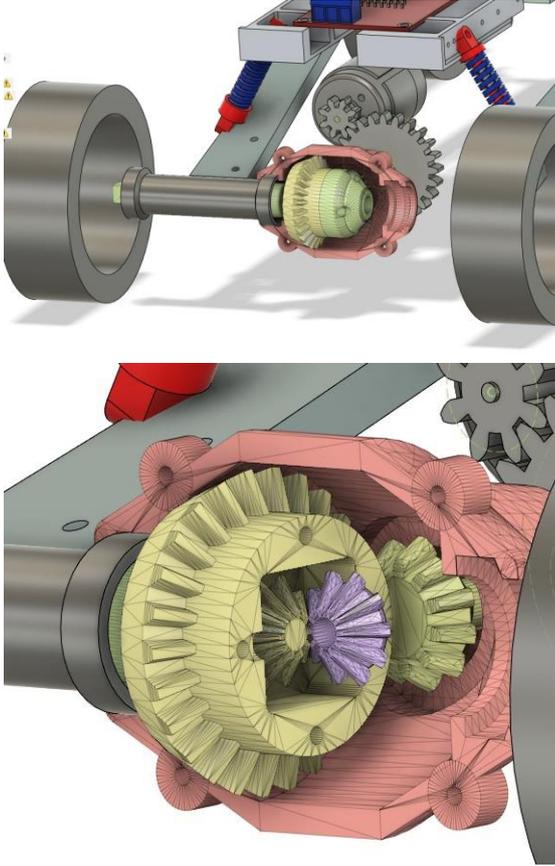
Обоснование актуальности проекта:

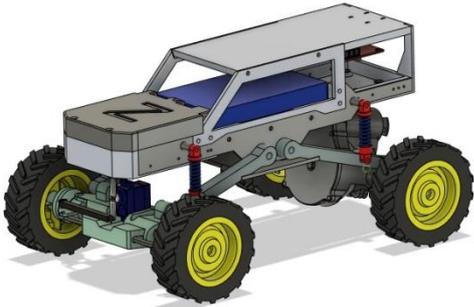
В современных условиях ведения боевых и разведывательных, оказание помощи по доставке медикаментов на простреливаемых участках, когда человек крайне опасно под пристальным контролем «птичек» преодолеть определенную дистанцию, то эту работу поможет выполнить собранный нами агрегат, тут не малую роль играют современные технологии. Широко используются летательные аппараты и механизированные роботы (на примере СВО). Возможность использования 3d моделирования для производства основных агрегатов и запчастей снижает себестоимость модели. Использование 3D печати непосредственно на месте так же снижает себестоимость готового изделия, что исключительно важно в подобных ситуациях. Использование готовых 3D моделей, представленных в интернет-сети бесплатно для компоновки автомодели - «доставка пакета первой помощи», с возможностью установки камеры, считаю актуальной.

Цель проекта:

1. Адаптировать 3D модели агрегатов к нашей модели;
2. по необходимости произвести корректировку под имеющиеся запчасти (оси, подшипники);
3. адаптировать прошивку arduino;
4. распечатать и протестировать полученные образцы.

Ход работы:

1	Изучение и компоновка основных узлов.	
2	Компоновка основных агрегатов.	
3	Адаптация размеров под имеющиеся детали и запчасти	

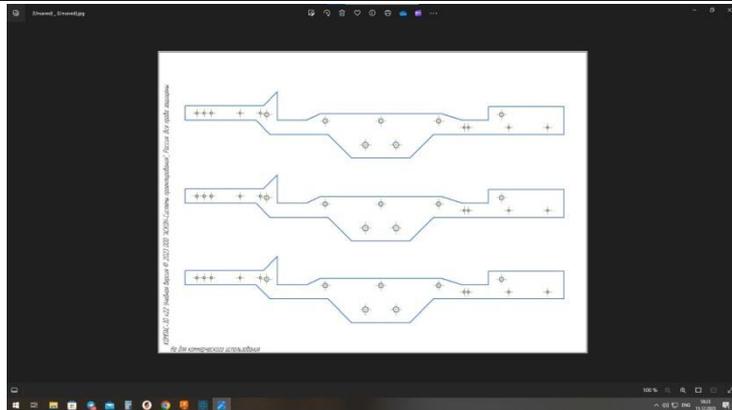
4	<p>Изменение кузова и крепления колес (подобран подходящий вариант с готовой модели), не исключая возможность самостоятельной печати колес</p>	
5	<p>Печать деталей на 3D принтере</p>	
6	<p>Установка осей дифференциала</p>	

7	Детали моста	 Two metal axle assemblies are shown side-by-side on a dark grey surface. Each assembly consists of a central shaft with a gear mounted on it, and two bearings or bushings positioned on either side of the gear. The background shows a blurred keyboard.
8	Корпус дифференциала	 A metal differential housing is shown from a top-down perspective on a dark grey surface. The housing is circular with a central opening and several mounting points around the perimeter. The background shows a blurred keyboard.
9	Тестовая сборка	 A complete axle assembly is shown on a dark grey surface. It features a central shaft with a gear in the middle, and two large, treaded tires mounted on the ends. The tires have yellow inner rims. The background shows a blurred keyboard with purple backlighting.

10 Работа с напечатанными деталями



11 Чертеж рамы для изготовления из металла

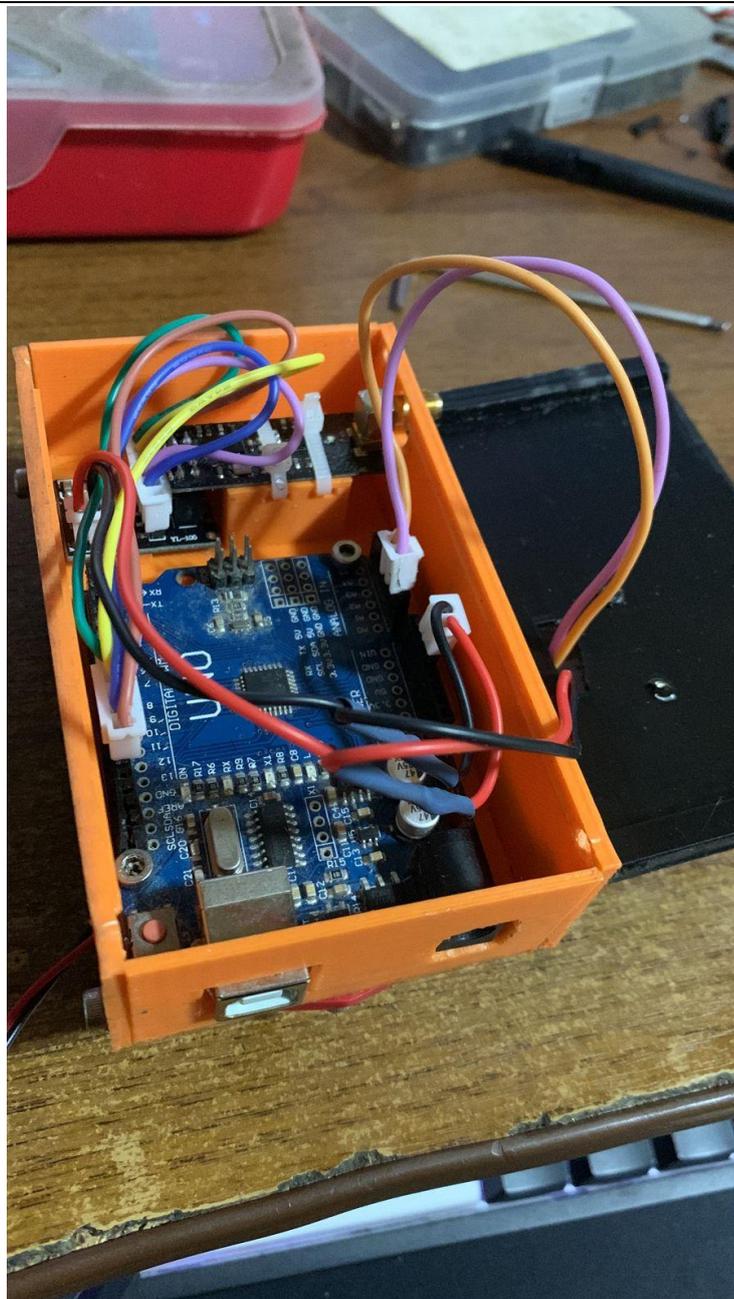


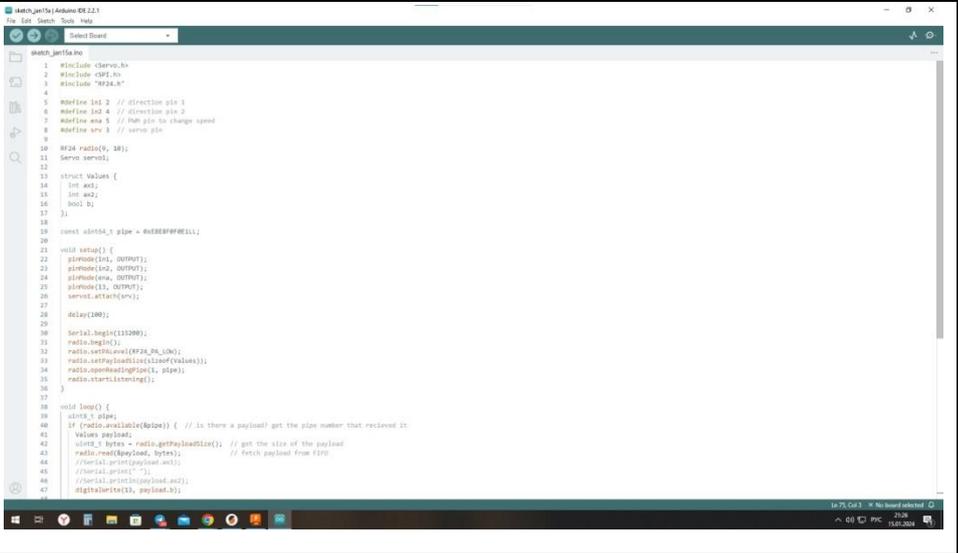
12	Сборка модели, вид спереди	 A photograph of a custom-built LEGO Technic car model. The car is primarily white and black, with large black tires mounted on yellow hubs. It features a white front bumper and a blue motor at the rear. The car is positioned on a light-colored tiled floor.
13	Модель, вид сзади	 A photograph of the same LEGO Technic car model from a rear perspective. This view highlights the blue motor, the black chassis, and the white frame. The large black tires with yellow hubs are clearly visible.
14	Работа с кодом на пульте	 A photograph showing a person from behind, sitting at a desk and working on a computer. The person's hands are on a keyboard with purple backlighting. The computer monitor displays a code editor interface with lines of text. On the desk, there are various electronic components, including a breadboard and several wires.

15 Монтаж пульта



16 Напечатанный корпус пульта управления



17	Собранный пульт	
18	Скеч ардуино	 <pre> sketch_jan15a.ino 1 #include <SPI.h> 2 #include <RF24.h> 3 #include "RF24.h" 4 5 #define led1 2 // direction pin 1 6 #define led2 4 // direction pin 2 7 #define swa 5 // SW pin to change speed 8 #define swb 3 // servo pin 9 10 RF24 radio(9, 10); 11 12 Servo servo; 13 14 struct Values { 15 int sw1; 16 int sw2; 17 bool b; 18 }; 19 20 const uint8_t pipe = 0x0000000000000000; 21 22 void setup() { 23 pinMode(led1, OUTPUT); 24 pinMode(led2, OUTPUT); 25 pinMode(swa, OUTPUT); 26 pinMode(swb, OUTPUT); 27 servo.attach(swb); 28 delay(1000); 29 Serial.begin(115200); 30 radio.begin(); 31 radio.setChannel(RF24_PA_LMS); 32 radio.setPayloadSize(sizeof(Values)); 33 radio.setPayloadLine(sizeof(Values)); 34 radio.openReadingPipe(1, pipe); 35 radio.startListening(); 36 } 37 38 void loop() { 39 if (radio.available(&pipe)) { // is there a payload? get the pipe number that received it. 40 Values payload; 41 uint8_t bytes = radio.getPayloadLine(); // get the size of the payload 42 radio.read(&payload, bytes); // fetch payload from FIFO 43 //Serial.println(payload.sw1); 44 //Serial.println(" "); 45 //Serial.println(payload.sw2); 46 digitalWrite(led1, payload.b); 47 } 48 } </pre>
19	Код пульта	<pre> #include <SPI.h> #include "RF24.h" RF24 radio(9,10); </pre>

```

#define potpin_1 A1
#define potpin_2 A0
#define potbut 6

struct Values {
    int ax1;
    int ax2;
    bool b;
};

const uint64_t pipe = 0xE8E8F0F0E1LL;

void setup() {
    pinMode(potbut, INPUT_PULLUP);
    radio.begin();
    radio.setPALevel(RF24_PA_LOW);
    radio.setPayloadSize(sizeof(Values));
    radio.openWritingPipe(pipe);
}

void loop() {
    Values payload;
    int a1, a2;
    bool button;

    a1 = 0;
    a2 = 0;

    for(int i=0; i<10; i++){
        a1 += analogRead(potpin_1);

```

		<pre> a2 += analogRead(potpin_2); } button = digitalRead(potbut); payload = { a1/10, a2/10, button }; radio.write(&payload, sizeof(Values)); } </pre>
20	Код машинки	<pre> #include <Servo.h> #include <SPI.h> #include "RF24.h" #define in1 2 // direction pin 1 #define in2 4 // direction pin 2 #define ena 5 // PWM pin to change speed #define srv 3 // servo pin RF24 radio(9, 10); Servo servo1; struct Values { int ax1; int ax2; bool b; }; const uint64_t pipe = 0xE8E8F0F0E1LL; void setup() { pinMode(in1, OUTPUT); </pre>

```

pinMode(in2, OUTPUT);
pinMode(ena, OUTPUT);
pinMode(13, OUTPUT);
servo1.attach(srv);

delay(100);

Serial.begin(115200);
radio.begin();
radio.setPALevel(RF24_PA_LOW);
radio.setPayloadSize(sizeof(Values));
radio.openReadingPipe(1, pipe);
radio.startListening();
}

void loop() {
  uint8_t pipe;
  if (radio.available(&pipe)) { // is there a payload? get the
    pipe number that recieved it
      Values payload;
      uint8_t bytes = radio.getPayloadSize(); // get the size of
the payload
      radio.read(&payload, bytes); // fetch payload from
FIFO
      //Serial.print(payload.ax1);
      //Serial.print(" ");
      //Serial.println(payload.ax2);
      digitalWrite(13, payload.b);

```

```

int pos;
pos = map(payload.ax1, 0, 1023, 7, 177);
servo1.write(pos);
int speed;
speed = payload.ax2;
Serial.print(speed);
Serial.print(" ");
const int hyst = 32;
if (speed - 1024/2 > hyst) {
  Serial.println("forw");
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  analogWrite(ena, map(speed, 1024 / 2 + hyst, 1023, 0,
255));
} else if (speed - 1024/2 < (-hyst)) {
  Serial.println("back");
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  analogWrite(ena, map(speed, 1024 / 2 - hyst, 0, 0, 255));
} else {
  Serial.println("stop");
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  analogWrite(ena, 0);
}
}
}

```

Фото готовой модели

