

Муниципальное бюджетное общеобразовательное учреждение
«Средняя общеобразовательная школа №1 имени Героя России С.А. Кислова»

ИНДИВИДУАЛЬНЫЙ ПРОЕКТ

Тип проекта: творческий

Тема: «*Грузовой автомобиль*»

Автор проекта:
Меньшенин Макар 5В класс
Наставник проекта:
учитель информатики
Невзорова Е.Н.

2024-2025 учебный год

Оглавление

Идея проекта	3
Актуальность проекта	3
Задачи проекта	3
Ход работы	4
Описание процесса моделирования.....	4
Описание процесса подготовки и изготовления модели	5
Программирование работы «Грузового автомобиля»	11
Ресурсы	15
Дальнейшее развитие проекта.....	15
Список используемой литературы.....	15

Идея проекта

Грузовой автомобиль - автомобиль, предназначенный главным образом для перевозки грузов или специального оборудования.

Основные направления совершенствования конструкции грузовых автомобилей – повышение их грузоподъёмности и грузовместимости, снижение токсичности и эксплуатационных затрат, внедрение систем повышения безопасности и облегчение условий труда водителя.

Для доставки партий разнообразных грузов используются различные виды транспорта, каждый из которых обладает определенными выгодными особенностями. Сложно переоценить значение грузовых перевозок, позволяющих успешно решать с помощью автотранспорта тысячи практических задач разного масштаба, степени сложности. Несмотря на большие объемы готовой продукции, сырья, материалов, перевозимых по железной дороге, морям, рекам, автомобили обладают неоспоримыми преимуществами. Используя грузовые машины, вы получаете удачную возможность доставить грузы из одной точки в другую без необходимости перегрузок.

Меня заинтересовала эта тема, т.к. в нашем городе очень часто можно видеть такие грузовики, обеспечивающие супермаркеты и другие магазины.

Занимаясь в кружке по робототехнике, я подумал, а почему бы не сконструировать грузовой автомобиль. Разобраться в деталях его конструкции. Проанализировать, какие типы конструкций можно воплотить в модели.

Актуальность проекта

Востребованность среди производителей, предпринимателей автотранспорта для грузоперевозок объясняется маневренностью, возможностью найти подходящую технику. Для разных задач можно подобрать не только тентовые, но обеспечивающие особый температурный режим рефрижераторные полуприцепы. Область применения максимально обширна, особенно при работе на дальних расстояниях на магистральных трассах. При этом они демонстрируют высокий уровень надежности, топливной экономичности и производительности. Расширение возможностей грузового транспорта – актуальная проблема современности.

Задачи проекта

- ✓ Изучить виды грузовых автомобилей;
- ✓ Сгенерировать ряд идей по созданию будущей модели;
- ✓ Создать модель «Грузовой автомобиль»;
- ✓ Создать программу для демонстрации возможностей модели.

Ход работы

Описание процесса моделирования

Изучив виды грузовых автомобилей, которые используются в настоящее время, я проанализировал данную ситуацию и у меня появился ряд идей по созданию будущей модели. Моя модель должна перемещаться (вперёд, назад, делать поворот), иметь удобную конструкцию для загрузки и выгрузки товаров.

Затем приступил к моделированию моей модели.

В моей модели будет использован электронный конструктор СМАРТ РОБО. «Смарт РОБО» - это конструктор, который позволяет собрать четырехколесного робота на базе платформы Arduino.



Рис.1

Для создания прицепа будем использовать детали конструктора Lego Mindstorms EV3. (Рис.2)



Рис.2

Описание процесса подготовки и изготовления модели

Изучив способы соединения и крепления деталей конструктора СМАРТ РОБО, я проанализировал данную ситуацию, и приступил к моделированию «Грузового автомобиля».

Сначала собрал четырехколесного робота на базе платформы Arduino. (рис.3).

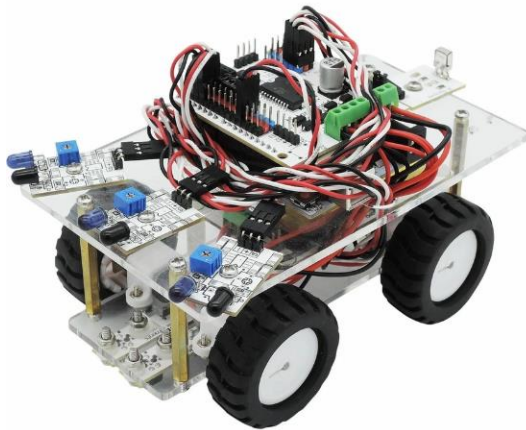


Рис.3

Усилил колёса, добавил дополнительно моторы.

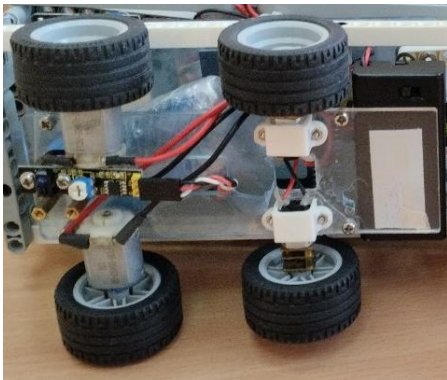


Рис. 4

Продумал соединительный крепёж для удлинения рамы под грузовой отсек.

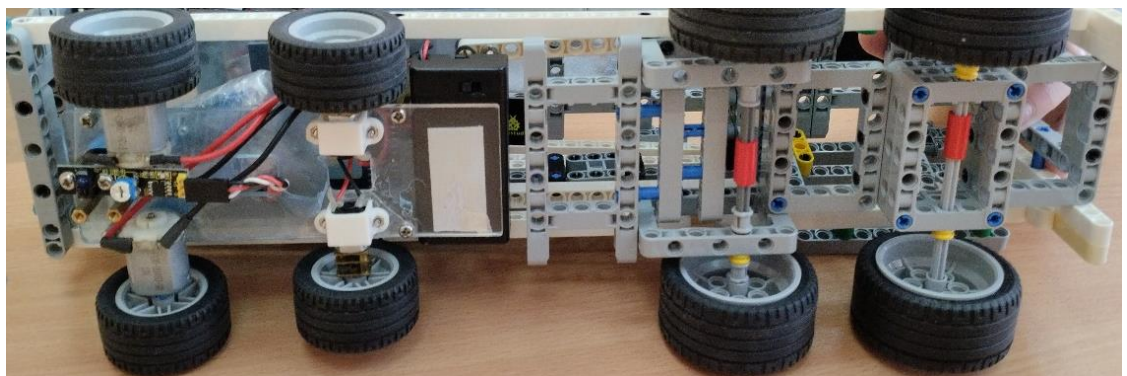


Рис. 5

Приступил к проектированию кузова грузового автомобиля.

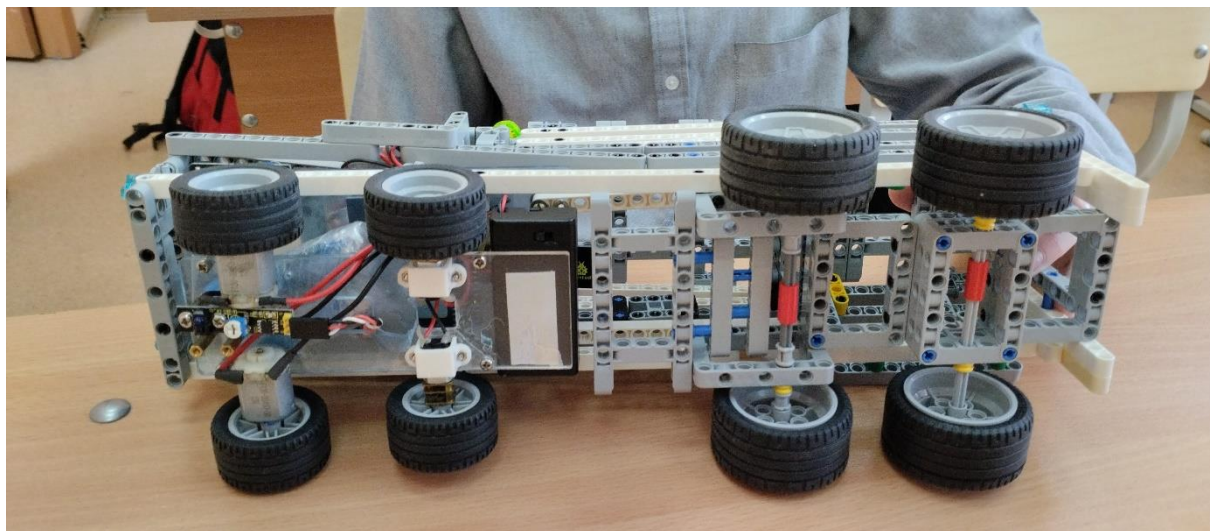


Рис. 6

Работал над деталями конструкции автомобиля, используя дополнительные элементы конструктора Lego Mindstorms EV3.

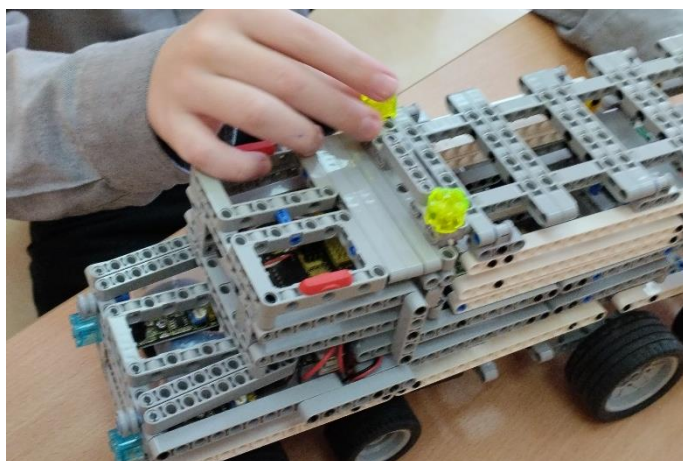


Рис.7

Усилил конструкцию (рис.8)

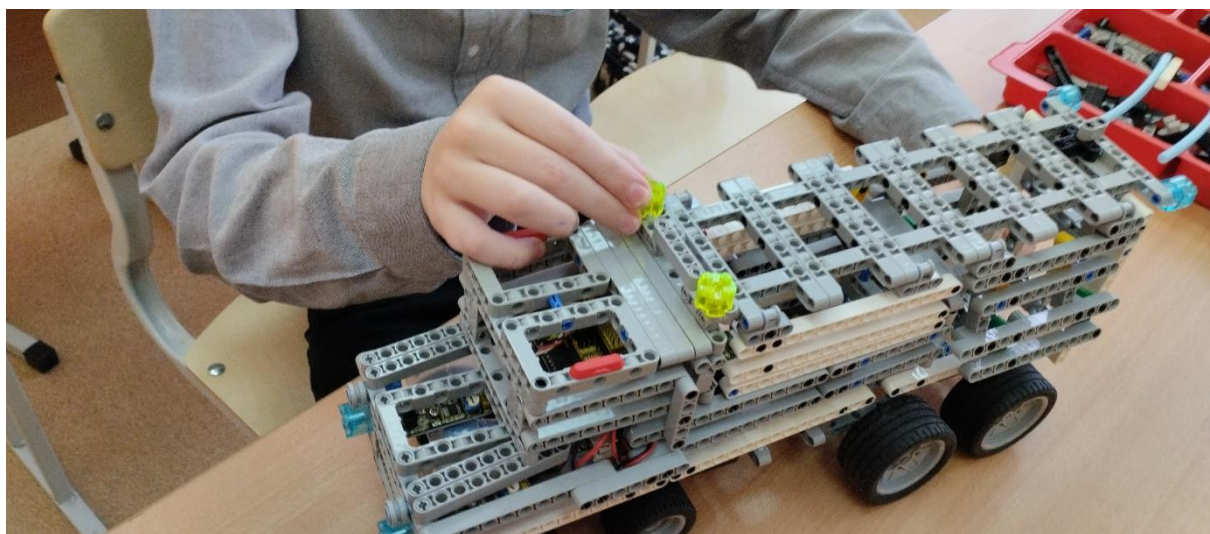


Рис. 8

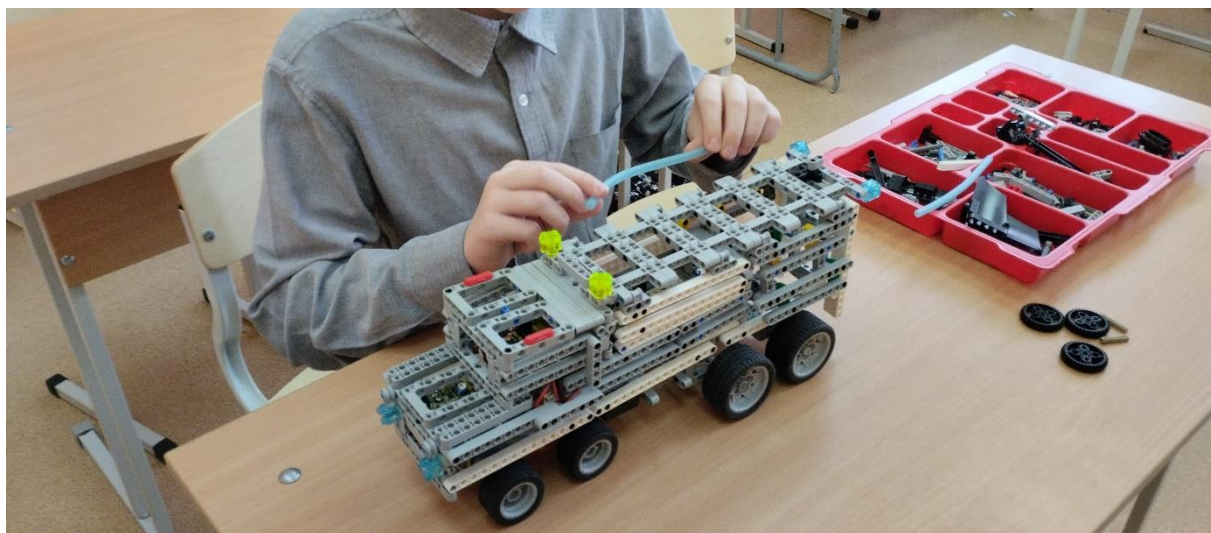


Рис. 9

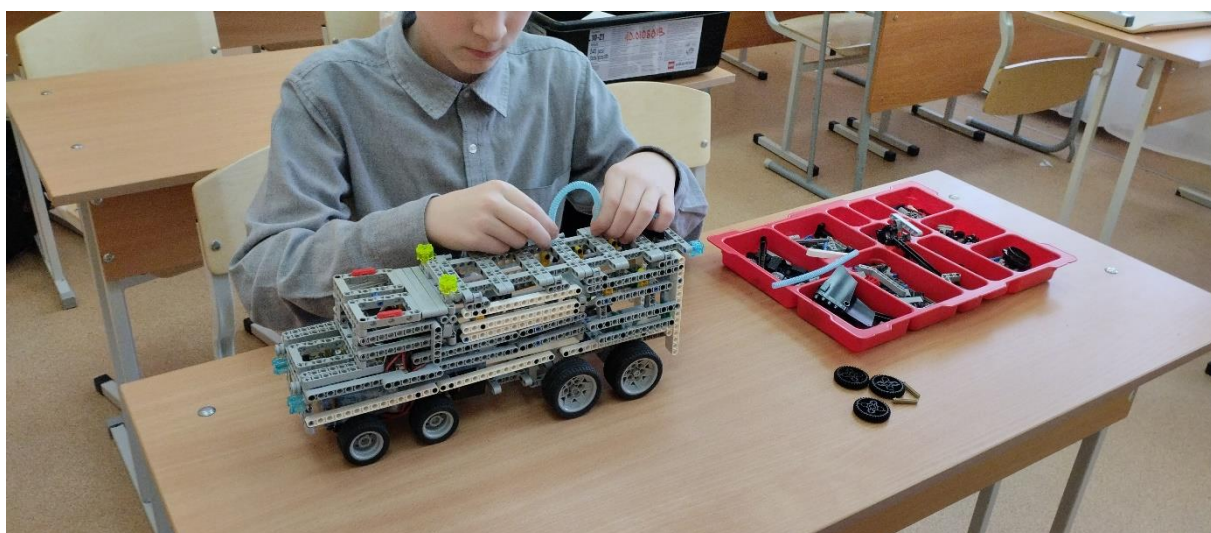


Рис.10

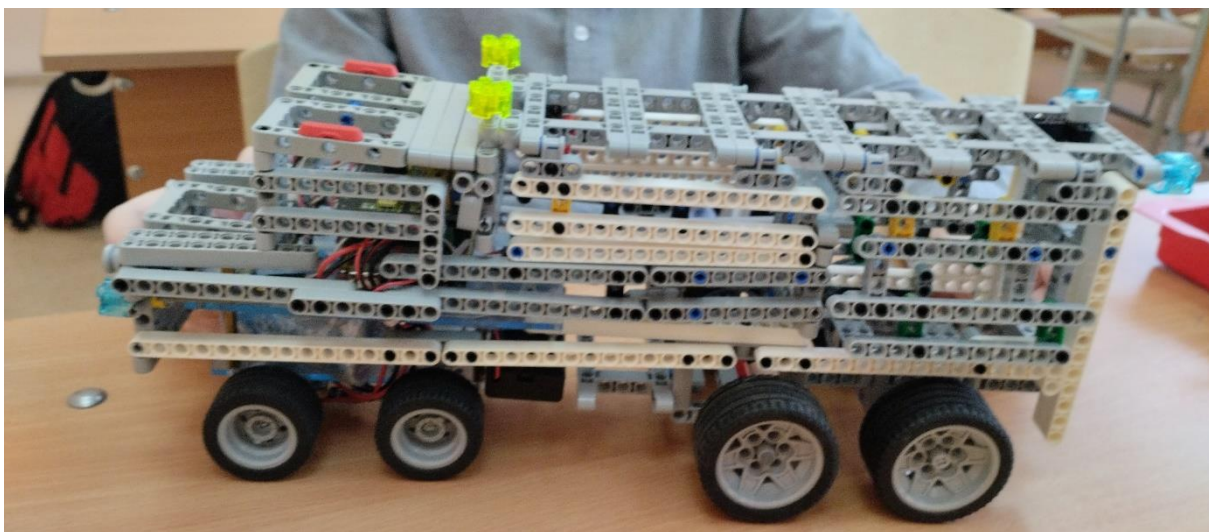


Рис.11



Рис. 12

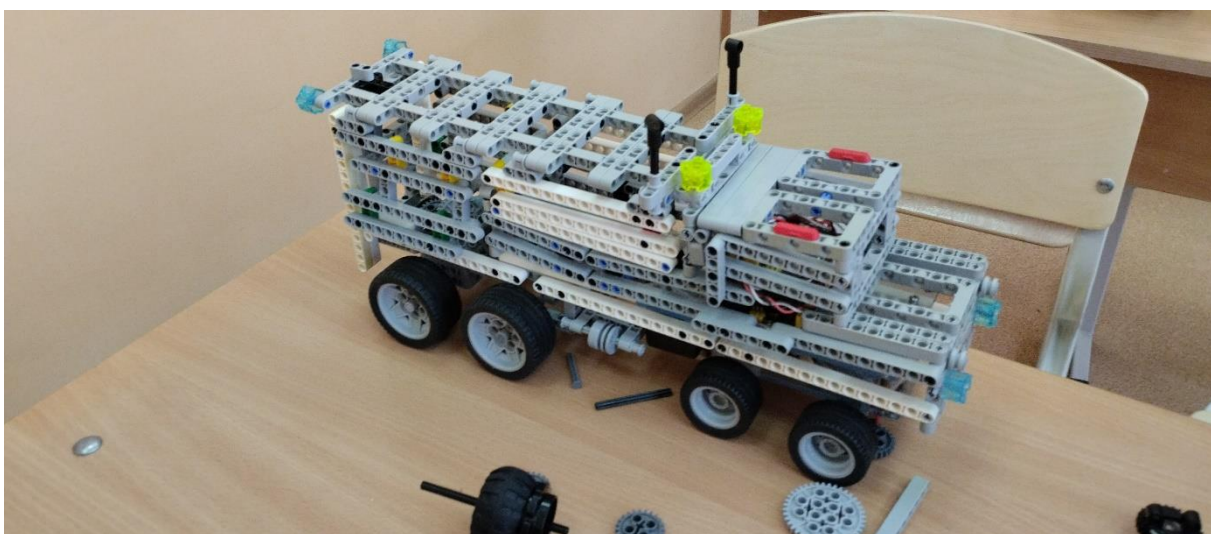


Рис.13



Рис.14

Собрал удобную конструкцию для загрузки и выгрузки товаров

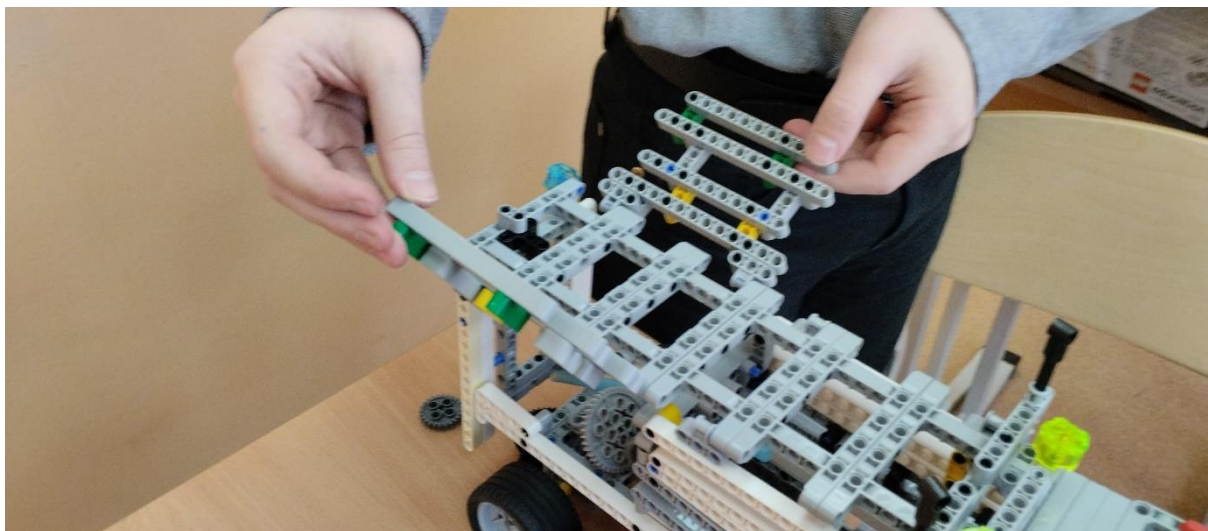


Рис.15

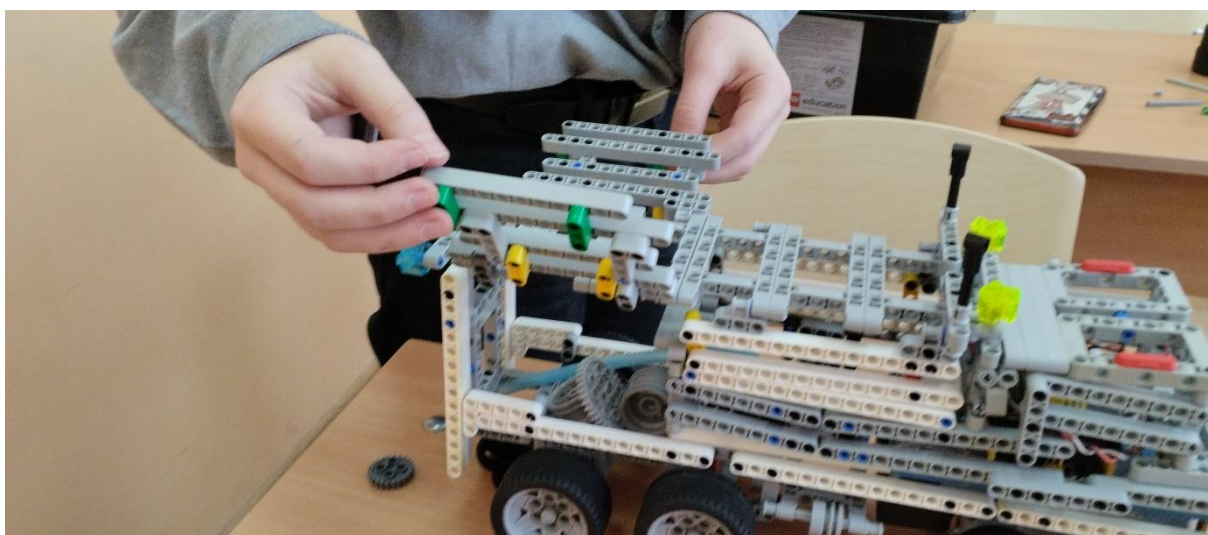


Рис.16



Рис.17

После того, как конструкция была готова, я приступил к оформлению. Подбирал детали конструктора для оформления внешнего вида.

Рис.18

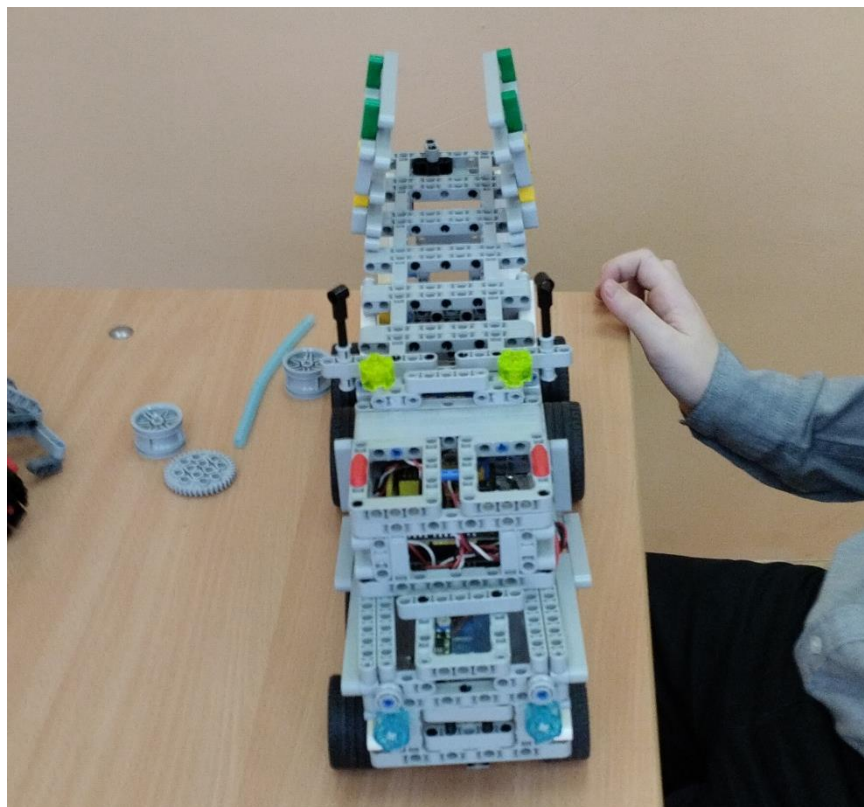
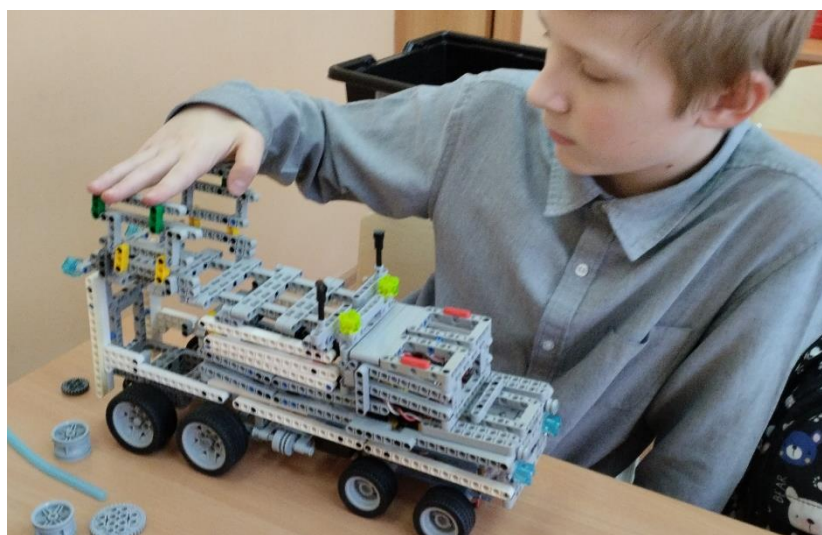


Рис.19



Программирование работы «Грузового автомобиля»

После того как конструкция была готова приступил к программированию работы моей модели. Разработчики позаботились о программировании электронного конструктора СМАРТ РОБО. На сайте производителя можно ознакомиться с кодами и библиотеками для набора. (Режим движения по линии, Режим объезда препятствий, Режим дистанционного управления). Разработчики использовали язык программирования C++

```
#include <IRremote.h> //Подключение библиотеки для получения и
обработки сигнала с ИК-приемника.

#define E1 6 //Вывод платы расширения Motor Drive Shield, используемый для
контроля скорости мотора А.
#define E2 5 //Вывод платы расширения Motor Drive Shield, используемый для
контроля скорости мотора В.
#define M1 7 //Вывод платы расширения Motor Drive Shield, используемый для
контроля направления вращения мотора А.
#define M2 4 //Вывод платы расширения Motor Drive Shield, используемый для
контроля направления вращения мотора В.

int motorSpeed = 255; //Переменная для хранения скорости вращения моторов
(вы можете задать свою скорость вращения колес).
int IR_pin = 10; //Вывод, к которому подключен ИК-приемник.
decode_results results; //Переменная для хранения принятых данных (кодов
нажатых кнопок на пульте).

IRrecv irreceiver(IR_pin); //Создание объекта приемника, имеющего
специфичные для приемника свойства и методы.
//Например, метод resume() возобновляет получение данных приемником после
декодирования результатов.

void motor_drive_config() //Инициализация выводов платы расширения Motor
Drive Shield.
{
    pinMode(M1, OUTPUT); //Назначение вывода M1 выходом.
    pinMode(M2, OUTPUT); //Назначение вывода M2 выходом.
    pinMode(E1, OUTPUT); //Назначение вывода E1 выходом.
    pinMode(E2, OUTPUT); //Назначение вывода E2 выходом.
}

void move_forward() //Движение робота вперед.
{
    digitalWrite(M1, HIGH); //Правые колеса вращаются вперед.
    digitalWrite(M2, HIGH); //Левые колеса вращаются вперед.
    analogWrite(E1, motorSpeed); //Скорость вращения правых колес.
    analogWrite(E2, motorSpeed); //Скорость вращения левых колес.
}

void move_right() //Движение робота вправо.
{
    digitalWrite(M1, LOW); //Правые колеса вращаются назад.
    digitalWrite(M2, HIGH); //Левые колеса вращаются вперед.
    analogWrite(E1, motorSpeed); //Скорость вращения левых колес.
    analogWrite(E2, motorSpeed); //Скорость вращения правых колес.
}

void move_left() //Движение робота влево.
{

```

```

    digitalWrite(M1, HIGH); //Правые колеса вращаются вперед.
    digitalWrite(M2, LOW); //Левые колеса вращаются назад.
    analogWrite(E1, motorSpeed); //Скорость вращения левых колес.
    analogWrite(E2, motorSpeed); //Скорость вращения правых колес.
}

void move_stop() //Остановка робота.
{
    digitalWrite(M1, LOW); //Правые колеса вращаются назад.
    digitalWrite(M2, LOW); //Левые колеса вращаются назад.
    analogWrite(E1, 0); //Скорость вращения колес снизить до 0.
    analogWrite(E2, 0); //Скорость вращения колес снизить до 0.
}

void move_back() //Движение робота назад.
{
    digitalWrite(M1, LOW); //Правые колеса вращаются назад.
    digitalWrite(M2, LOW); //Левые колеса вращаются назад.
    analogWrite(E1, motorSpeed); //Скорость вращения левых колес.
    analogWrite(E2, motorSpeed); //Скорость вращения правых колес.
}

void stop_reboot() //Остановка для передышки.
{
    move_stop(); //Остановка.
    delay(150); //Отдых 150 мс.
}

void line_moving(const boolean& fwd) //Движение по линии.
{
    if (fwd) { //Если первое движение вперед.
        move_forward(); //То двигаться вперед.
    } else { //Иначе.
        move_back(); //Двигаться назад.
    }

    delay(1500); //Время движения 1,5 сек.
    stop_reboot(); //Остановка для передышки.

    if (fwd) { //Если первое движение было вперед.
        move_back(); //То двигаться назад.
    } else { //Иначе.
        move_forward(); //Двигаться вперед.
    }

    delay(1500); //Время движения 1,5 сек.
    move_stop(); //Остановиться.
}

void turn_right() //Поворот вправо.
{
    move_right(); //Движение вправо.
    delay(770); //Время движения 770 мс.
}

void turn_left() //Поворот влево.
{
    move_left(); //Двигаться влево.
    delay(820); //Время движения 820 мс.
}

void straight_step() //Шаг прямо.
{

```

```

    move_forward(); //Двигаться прямо.
    delay(1500); //Время движения прямо.
}

void square_moving(const boolean& right) //Движение по квадрату.
{
    int i = 4; //Количество шагов равное 4.
    while (i > 0) { //Цикл по шагам.
        straight_step(); //Совершить шаг вперёд.
        if (right) { //Если квадрат правый.
            turn_right(); //Поворот вправо.
        } else { //Иначе.
            turn_left(); //Поворот влево.
        }
        --i; //Уменьшить количество шагов.
    }
    move_stop(); //Остановиться.
}

void corner_moving(const boolean& right) //Движение "угол".
{
    if (right) { //Если угол правый.
        turn_right(); //Повернуть вправо.
    } else { //Иначе.
        turn_left(); //Повернуть влево.
    }

    straight_step(); //Движение прямо.
    stop_reboot(); //Остановка для передышки.

    if (right) { //Если угол правый.
        turn_left(); //То второй поворот влево.
    } else { //Иначе.
        turn_right(); //Второй поворот вправо.
    }

    straight_step(); //Движение прямо.
    stop_reboot(); //Отдых.
}

void reverse(const boolean& right) //Разворот.
{
    if (right) { //Если разворот правый.
        turn_right(); //Повренуть вправо.
        turn_right(); //Повернуть вправо ещё раз.
    } else { //Иначе - он левый.
        turn_left(); //Повернуть влево.
        turn_left(); //Повернуть влево ещё раз.
    }
    stop_reboot(); //Остановка для отдыха.
}

void snake_mode() { //Режим змейки.
    corner_moving(true); //Движение по углу вправо.
    corner_moving(false); //Движение по углу влево.

    reverse(true); //Разворот вправо.

    corner_moving(false); //Движение по углу влево.
    corner_moving(true); //Движение по углу вправо.

    reverse(false); //Разворот влево.
}

```

```

void setup() //Задание начальных параметров системы.
{
    irreceiver.enableIRIn(); //Инициализация ИК-приемника.
    motor_drive_config(); //Инициализация выводов платы расширения Motor
    Drive Shield.
    move_stop(); //Начальная позиция робота — робот стоит.
}

void loop() //Основной цикл (алгоритм описан в разделе «Алгоритм работы
робота»).
{
    if (irreceiver.decode(&results)) { //Если приемник получил сигнал..

        if (results.value == 16724175) { //Если принятое значение равно коду
        кнопки 1...
            line_moving(true); //Движение вперед и назад.
        } else if (results.value == 16718055) { //Или, если значение равно коду
        кнопки 2...
            line_moving(false); //Движение назад и вперед.
        } else if (results.value == 16743045) { //Или, если значение равно коду
        кнопки 3...
            square_moving(true); //Квадрат с поворотом вправо.
        } else if (results.value == 16716015) { //Или, если значение равно коду
        кнопки 4...
            square_moving(false); //Квадрат с поворотом влево.
        } else if (results.value == 16726215) { //Или, если значение равно коду
        кнопки 5...
            corner_moving(true); //Угол с поворотом вправо.
        } else if (results.value == 16734885) { //Или, если значение равно коду
        кнопки 6...
            corner_moving(false); //Угол с поворотом влево.
        } else if (results.value == 16728765) { //Или, если значение равно коду
        кнопки 7...
            reverse(true); //Разворот вправо.
        } else if (results.value == 16730805) { //Или, если значение равно коду
        кнопки 8...
            reverse(false); //Разворот влево.
        } else if (results.value == 16732845) { //Или, если значение равно коду
        кнопки 9...
            snake_mode(); //Змейка.
        }

        irreceiver.resume(); //Возвращаем ИК-приемник в режим ожидания.
    }
}

```

Многokrатно тестировал моего робота на движение.

По данной программе моя модель может: двигаться вперёд, назад, объезжать препятствия.

После того как всё получилось, заснял работу моей модели на видеокамеру и создал видеофильм.

Ссылка на видеофильм: <https://disk.yandex.ru/i/eIoWEuarcHfXGQ>

Ресурсы

- ✓ Конструктор СМАРТ РОБО, LEGO Education Mindstorms EV3 расширенный набор 45560
- ✓ Программное обеспечение: язык программирования C++

Дальнейшее развитие проекта

Практическая значимость проекта заключается в том, что проектный продукт можно будет использовать в учебном процессе (классные часы, уроки информатики, физики, предметные недели) для популяризации технического творчества. В дальнейшем можно заняться дистанционным управлением данного робота.

Список используемой литературы

1. Научись программировать. - [Электронный ресурс]. URL: <https://www.lego.com/ru-ru/themes/mindstorms/learntoprogram>
(дата обращения 27.01.2025)
2. Программирование роботов: методы и инструменты. - [Электронный ресурс]. URL: <https://gb.ru/blog/programmirovanie-robotov/>
(дата обращения 02.02.2025)
3. Коды и библиотеки для набора «Смарт РОБО». - [Электронный ресурс]. URL: <https://smartelements.ru/page/robokit> (07.02.2025)